

# **Phase 3 Project:**

## **Implement Spring Security with Authentication**

**GitHub Link :**

<https://github.com/Karthiga-web/SpringSecurityProject.git>

**Algorithm:**

1. A spring boot application is created with Controller, JpaRepository, Entity class for User and Role, application properties file, JSP
2. The Crud Repository class is used to call methods from Repository for getting information about data from the database.
3. The Entity class links the Database table with the form the feedback is given to the controller
4. The controller redirects the link from jsp to service and repository for data validation
5. WebSecurityConfig decides the permission allocation to the links with overridden methods like configure
6. It has methods that check if data is present in DB and saves the data or updates the data to DB
7. The RequestMapping annotation get and post the data from/to the output page
8. The output starts with the Welcome page where the user is prompted to click a link
9. Enter credentials page occurs where username and password is given by the user
10. If correct credentials are given, the hello page is displayed for Admin and hellouser page is displayed for the user
11. If irrespective links are loaded, then it leads to 403 custom error handling page.

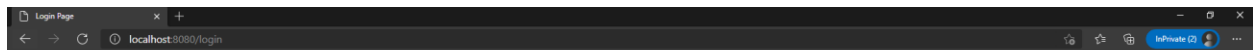
12. A sign out option leads to logging out of the application leading to the enter credentials page
13. The page loads only for the correct role and error page occurs for irrespective roles

## Outputs:



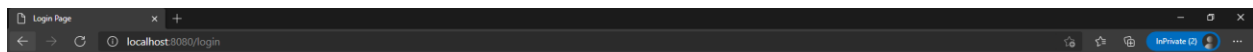
Welcome!

Click [here](#) to see a greeting.



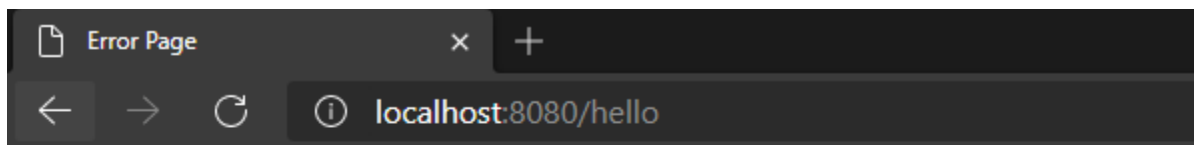
Enter your Credentials!

User Name
Password
Sign In



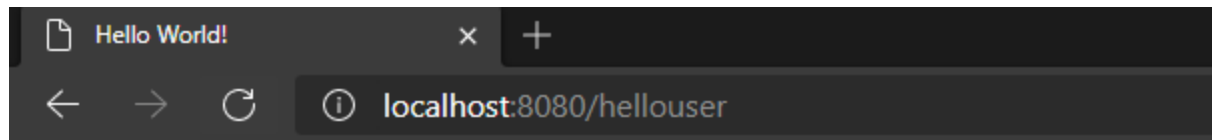
Enter your Credentials!

karthi
***
Sign In



**You do not have permission to access this page!**

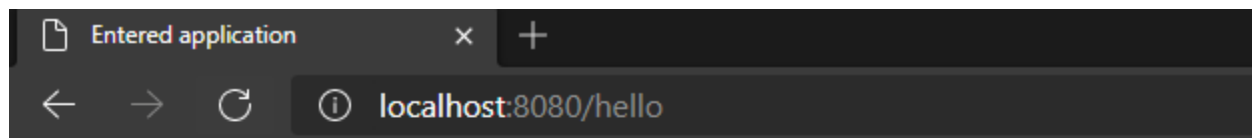
Sign in as different user



**Entered into Application of User Access!**

**Hello karthi**

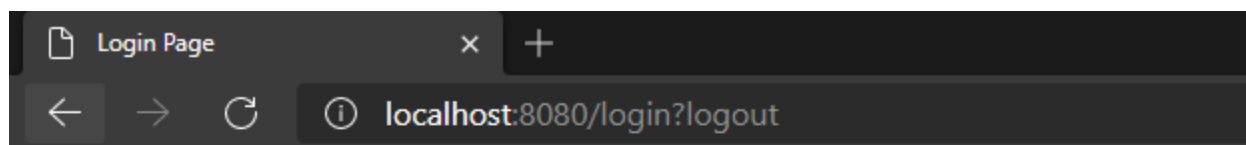
Sign Out



**Entered into Application of Admin Access!**

**Hello laks**

Sign Out



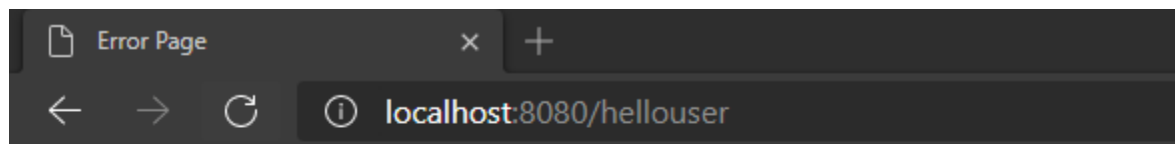
**Enter your Credentials!**

laks

....

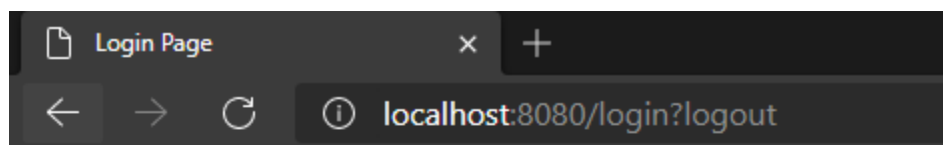
Sign In

You have been logged out.



**You do not have permission to access this page!**

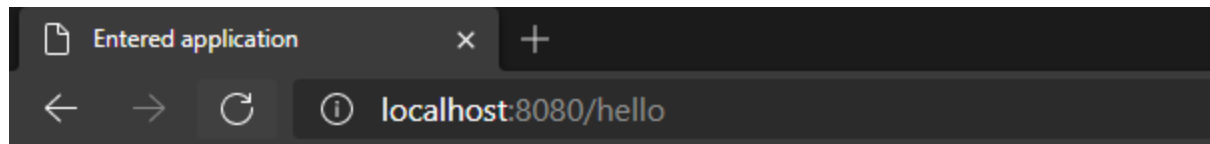
[Sign in as different user](#)



**Enter your Credentials!**

[Sign In](#)

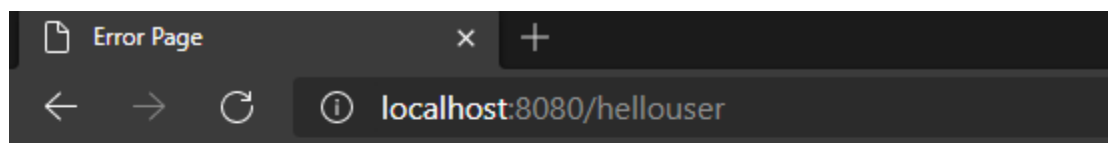
You have been logged out.



**Entered into Application of Admin Access!**

**Hello ramk**

[Sign Out](#)



**You do not have permission to access this page!**

[Sign in as different user](#)

**Database:**

Navigator

SCHEMAS

Filter objects

karthiga

drinks

hibernate\_sequence

role

Columns

Indexes

Foreign Keys

Triggers

user

Columns

Indexes

Foreign Keys

Triggers

user\_role

Columns

Indexes

Foreign Keys

Triggers

Views

Administration

Schemas

Information

Schema: karthiga

Navigator

SCHEMAS

Filter objects

karthiga

drinks

hibernate\_sequence

role

Columns

Indexes

Foreign Keys

Triggers

user

Columns

Indexes

Foreign Keys

Triggers

user\_role

Columns

Indexes

Foreign Keys

Triggers

Views

Administration

Schemas

Information

Schema: karthiga

Navigator

SCHEMAS

Filter objects

karthiga

drinks

hibernate\_sequence

role

Columns

Indexes

Foreign Keys

Triggers

user

Columns

Indexes

Foreign Keys

Triggers

user\_role

Columns

Indexes

Foreign Keys

Triggers

Views

Administration

Schemas

Information

Schema: karthiga

role

user

user\_role

Limit to 1000 rows

1 • SELECT \* FROM karthiga.user;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

Result Grid

Form Editor

Field Types

id	email	firstname	lastname	password	role	username
1	myemail@gmail.com	karthiga	gr	\$2a\$10\$to5a4QdbMku.YXffizq8Bu9pW5yeZTAhx8pmaxiCgeUhQ9N7zuYJC	ROLE_USER	karthi
2	myemail@gmail.com	lakshmi	devi	\$2a\$10\$TTrdZ2tZom3vCznazmQ9/Od8LT4fblokinr0V888f7gPMFOf7IW	ROLE_ADMIN	laks
3	nomail@gmail.com	ram	kumar	\$2a\$10\$S0DfP9v8vv.UXAi5K8jIdOLNbVY0D2xE8ws.45P2WGOHgwGbFEIF.	ROLE_ADMIN	ramik
NULL	NULL	NULL	NULL	NULL	NULL	NULL

user 1 ×

role

user

user\_role

role

Limit to 1000 rows

1 • SELECT \* FROM karthiga.role;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

Result Grid

Form Editor

Field Types

id	name
1	ROLE_ADMIN
2	ROLE_USER
NULL	NULL

role 1 ×

role

user

user\_role

role

user\_role

Limit to 1000 rows

1 • SELECT \* FROM karthiga.user\_role;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

Result Grid

Form Editor

Field Types

user_id	role_id
2	1
3	1
1	2
NULL	NULL

user\_role 1 ×

**Code:**

RoleRepository.java X

```
1 package com.hcl.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5
6 @Repository
7 public interface RoleRepository extends JpaRepository<Role, Long>{
8
9 }
10
11
12
```

WebSecurityConfig.java


JunitController.java

JunitEntity.java

Role.java X

```
1 package com.hcl.entity;
2
3 import java.util.Set;
4
5 @Entity
6 @Table(name = "role")
7 public class Role {
8
9     @Id
10     @GeneratedValue(strategy = GenerationType.AUTO)
11     @Column(name = "id")
12     private Long id;
13
14     @Column(name = "name")
15     private String name;
16
17     @ManyToMany(mappedBy = "roles")
18     private Set<JunitEntity> users;
19
20     public Long getId() {
21         return id;
22     }
23
24     public void setId(Long id) {
25         this.id = id;
26     }
27
28     public String getName() {
29         return name;
30     }
31
32     public void setName(String name) {
33         this.name = name;
34     }
35
36     public Set<JunitEntity> getUsers() {
37         return users;
38     }
39
40     public void setUsers(Set<JunitEntity> users) {
41         this.users = users;
42     }
43 }
44
45
```



 application.properties X

```
1 spring.mvc.view.prefix = /WEB-INF/views/
2 spring.mvc.view.suffix = .jsp
3 spring.mvc.static-path-pattern=/resources/**
4 spring.jpa.hibernate.ddl-auto=update
5 spring.datasource.url= jdbc:mysql://localhost:3306/karthiga
6 spring.datasource.username=root
7 spring.datasource.password=root
8 spring.jpa.properties.hibernate.format_sql=true
9 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
10 logging.level.org.springframework.web: DEBUG
11 server.port=8080
```

```

home.jsp login.jsp SpringSecurityProject/pom.xml X
38     </dependency>
39     <dependency>
40         <groupId>mysql</groupId>
41         <artifactId>mysql-connector-java</artifactId>
42         <scope>runtime</scope>
43     </dependency>
44     <dependency>
45         <groupId>org.junit.jupiter</groupId>
46         <artifactId>junit-jupiter-api</artifactId>
47         <scope>test</scope>
48     </dependency>
49     <dependency>
50         <groupId>javax.servlet</groupId>
51         <artifactId>jstl</artifactId>
52     </dependency>
53     <dependency>
54         <groupId>org.springframework.boot</groupId>
55         <artifactId>spring-boot-starter-tomcat</artifactId>
56         <scope>provided</scope>
57     </dependency>
58     <dependency>
59         <groupId>org.springframework.security</groupId>
60         <artifactId>spring-security-test</artifactId>
61         <scope>test</scope>
62     </dependency>
63     <dependency>
64         <groupId>org.springframework.boot</groupId>
65         <artifactId>spring-boot-starter-data-jpa</artifactId>
66     </dependency>
67     <dependency>
68         <groupId>org.springframework.boot</groupId>
69         <artifactId>spring-boot-starter-test</artifactId>
70         <scope>test</scope>
71     </dependency>
72 </dependencies>
73 <build>
74     <plugins>
75         <plugin>
76             <groupId>org.springframework.boot</groupId>
77             <artifactId>spring-boot-maven-plugin</artifactId>
78         </plugin>
79         <plugin>
80             <groupId>org.apache.maven.plugins</groupId>
81             <artifactId>maven-surefire-plugin</artifactId>
82             <configuration>
83                 <testFailureIgnore>true</testFailureIgnore>
84             </configuration>
85         </plugin>
86     </plugins>
87 </build>
88 </project>
89

```

```
home.jsp x login.jsp SpringSecurityProject/pom.xml x
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <parent>
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-parent</artifactId>
9     <version>2.4.2</version>
10    <relativePath />
11  </parent>
12  <groupId>com.hcl</groupId>
13  <artifactId>SpringSecurityProject</artifactId>
14  <version>0.0.1-SNAPSHOT</version>
15  <packaging>war</packaging>
16  <name>SpringSecurityProject</name>
17  <description>Spring Security Project</description>
18  <properties>
19    <java.version>1.8</java.version>
20  </properties>
21  <dependencies>
22    <dependency>
23      <groupId>org.springframework.boot</groupId>
24      <artifactId>spring-boot-starter-data-jpa</artifactId>
25    </dependency>
26    <dependency>
27      <groupId>org.springframework.boot</groupId>
28      <artifactId>spring-boot-starter-web</artifactId>
29    </dependency>
30    <dependency>
31      <groupId>org.springframework.boot</groupId>
32      <artifactId>spring-boot-starter-security</artifactId>
33    </dependency>
34    <dependency>
35      <groupId>org.apache.tomcat.embed</groupId>
36      <artifactId>tomcat-embed-jasper</artifactId>
37      <scope>provided</scope>
38    </dependency>
39    <dependency>
40      <groupId>mysql</groupId>
41      <artifactId>mysql-connector-java</artifactId>
42      <scope>runtime</scope>
43    </dependency>
44    <dependency>
45      <groupId>org.junit.jupiter</groupId>
46      <artifactId>junit-jupiter-api</artifactId>
47      <scope>test</scope>
48    </dependency>
49    <dependency>
50      <groupId>javax.servlet</groupId>
51      <artifactId>jstl</artifactId>
52    </dependency>
```

```
home.jsp x login.jsp x
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
4 <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
5 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
6 <!DOCTYPE html>
7 <html>
8 <title>Login Page</title>
9 </head>
10 <body>
11 <h2>Enter your Credentials!</h2>
12 <form action="/login" method="post">
13 <div>
14 <input type="text" name="username" placeholder="User Name" />
15 </div>
16 <div>
17 <input type="password" name="password" placeholder="Password" />
18 </div>
19 <div>
20 <input type="submit" value="Sign In" />
21 </div>
22 <c:if test="${param.error ne null}">
23 <div>Invalid username and password.</div>
24 </c:if>
25 <c:if test="${param.logout ne null}">
26 <div>You have been logged out.</div>
27 </c:if>
28 <input type="hidden" name="${_csrf.parameterName}"
29   value="${_csrf.token}" />
30 </form>
31 </body>
32 </html>
33
```

```
home.jsp x
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
4 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
5 <!DOCTYPE html>
6 <html>
7 <head>
8 <title>Welcome to Application</title>
9 </head>
10 <body>
11 <h1>Welcome!</h1>
12 Click
13 <a href="<spring:url value='/hello' />">here</a> to see a greeting.
14 </body>
15 </html>
16
```

```

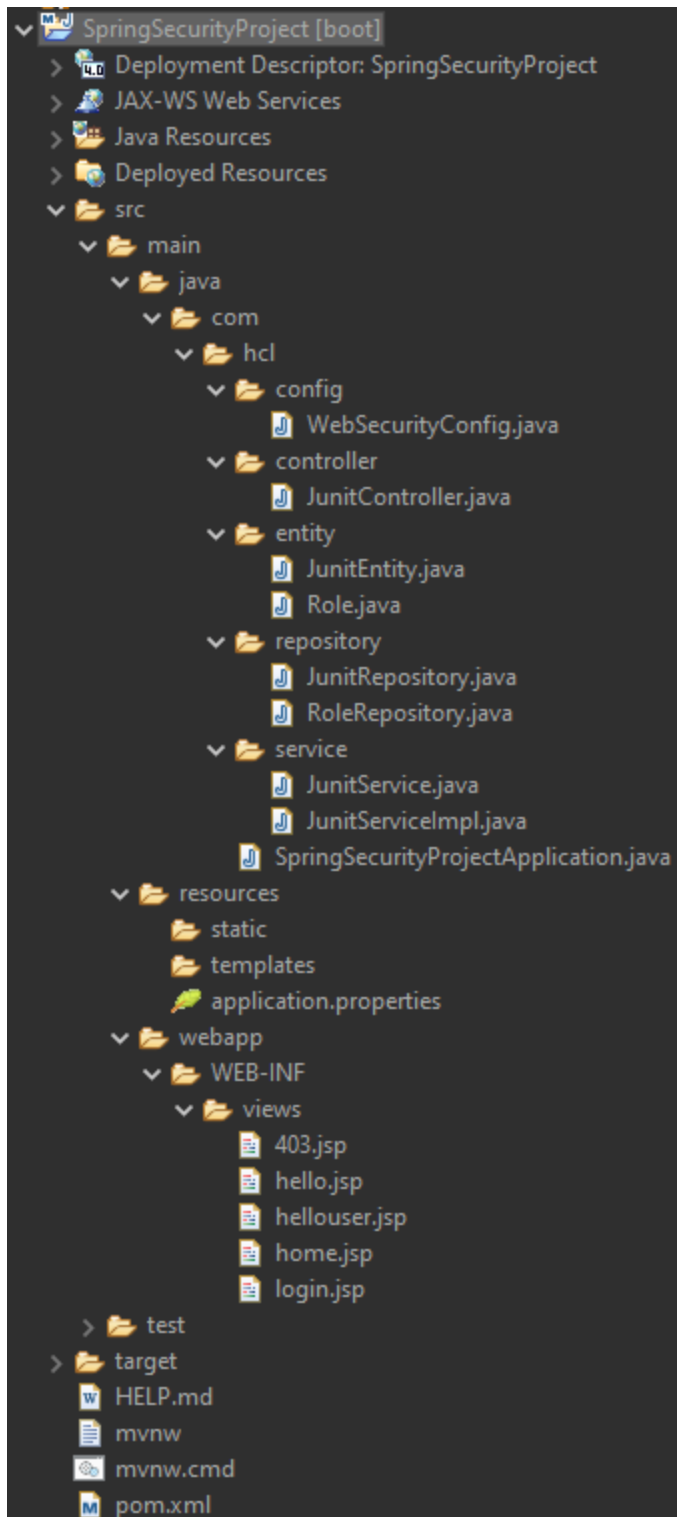
hello.jsp  hellouser.jsp x
1 |<%@ page language="java" contentType="text/html; charset=UTF-8"
2 |   pageEncoding="UTF-8"%>
3 |<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
4 |<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
5 |<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
6 |<!DOCTYPE html>
7 |<html>
8 |<head>
9 |  <title>Hello World!</title>
10|</head>
11|<body>
12|  <h2>Entered into Application of User Access!</h2>
13|  <h1>
14|    Hello <b><c:out value="${pageContext.request.remoteUser}"></c:out></b>
15|  </h1>
16|  <form action="/logout" method="post">
17|    <input type="submit" value="Sign Out" /> <input
18|      type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
19|  </form>
20|</body>
21|</html>
22|

```

```

hello.jsp x
1 |<%@ page language="java" contentType="text/html; charset=UTF-8"
2 |   pageEncoding="UTF-8"%>
3 |<%@ taglib prefix="spring" uri="http://www.springframework.org/tags"%>
4 |<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
5 |<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
6 |<!DOCTYPE html>
7 |<html>
8 |<head>
9 |  <title>Entered application</title>
10|</head>
11|<body>
12|  <h2>Entered into Application of Admin Access!</h2>
13|  <h1>
14|    Hello <b><c:out value="${pageContext.request.remoteUser}"></c:out></b>
15|  </h1>
16|  <form action="/logout" method="post">
17|    <input type="submit" value="Sign Out" /> <input
18|      type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
19|  </form>
20|</body>
21|</html>
22|

```



WebSecurityConfig.java JunitController.java JunitEntity.java X

```
75  
76 @Column(name = "email")  
77 String email;  
78 @Column(name = "role")  
79 String role;  
80  
81 public Integer getId() {  
82     return id;  
83 }  
84  
85 public void setId(Integer id) {  
86     this.id = id;  
87 }  
88  
89 public String getUsername() {  
90     return username;  
91 }  
92  
93 public void setUsername(String username) {  
94     this.username = username;  
95 }  
96  
97 public String getPassword() {  
98     return password;  
99 }  
100  
101 public void setPassword(String password) {  
102     this.password = password;  
103 }  
104  
105 public Set<Role> getRoles() {  
106     return roles;  
107 }  
108  
109 /**  
110  * @return the role  
111  */  
112 public String getRole() {  
113     return role;  
114 }  
115  
116 /**  
117  * @param role the role to set  
118  */  
119 public void setRole(String role) {  
120     this.role = role;  
121 }  
122  
123 public void setRoles(Set<Role> roles) {  
124     this.roles = roles;  
125 }  
126 }  
127
```

WebSecurityConfig.java X JunitController.java JunitEntity.java X

```
39     }
40
41     /**
42      * @param firstname the firstname to set
43      */
44     public void setFirstname(String firstname) {
45         this.firstname = firstname;
46     }
47
48     /**
49      * @return the lastname
50      */
51     public String getLastname() {
52         return lastname;
53     }
54
55     /**
56      * @param lastname the lastname to set
57      */
58     public void setLastname(String lastname) {
59         this.lastname = lastname;
60     }
61
62     /**
63      * @return the email
64      */
65     public String getEmail() {
66         return email;
67     }
68
69     /**
70      * @param email the email to set
71      */
72     public void setEmail(String email) {
73         this.email = email;
74     }
75
76     @Column(name = "email")
77     String email;
78     @Column(name = "role")
79     String role;
80
81     public Integer getId() {
82         return id;
83     }
84
85     public void setId(Integer id) {
86         this.id = id;
87     }
88
89     public String getUsername() {
90         return username;
91     }
```

<



```

WebSecurityConfig.java  JunitController.java  JunitEntity.java x
1 package com.hcl.entity;
2
3 import java.util.Set;
4
13
14 @Entity
15 @Table(name = "user")
16 public class JunitEntity {
17     @Id
18     @GeneratedValue(strategy = GenerationType.AUTO)
19     @Column(name = "id")
20     Integer id;
21     @ManyToMany
22     @JoinTable(name = "user_role", joinColumns = @JoinColumn(name = "user_id"), inverseJoinColumns = @JoinColumn(name = "role_id"))
23     private Set<Role> roles;
24
25     @Column(name = "username")
26     String username;
27     @Column(name = "password")
28     String password;
29     @Column(name = "firstname")
30     String firstname;
31     @Column(name = "lastname")
32     String lastname;
33
34     /**
35      * @return the firstname
36      */
37     public String getFirstname() {
38         return firstname;
39     }
40
41     /**
42      * @param firstname the firstname to set
43      */
44     public void setFirstname(String firstname) {
45         this.firstname = firstname;
46     }
47
48     /**
49      * @return the lastname
50      */
51     public String getLastName() {
52         return lastname;
53     }
54
55     /**
56      * @param lastname the lastname to set
57      */
58     public void setLastName(String lastname) {
59         this.lastname = lastname;
60     }
61
62     /**
63      *
64      */

```

```
WebSecurityConfig.java X JunitController.java X
1 package com.hcl.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
13
14 @Controller
15 public class JunitController {
16
17     @Autowired
18     JunitService service;
19
20     @RequestMapping(value= {"/","/home"})
21     public String mainhome()
22     {
23         return "home";
24     }
25     @RequestMapping("/hello")
26     public String hello()
27     {
28         return "hello";
29     }
30
31     @RequestMapping("/hellouser")
32     public String hellouser()
33     {
34         return "hellouser";
35     }
36     @RequestMapping("/login")
37     public String login()
38     {
39         return "login";
40     }
41     @RequestMapping("/403")
42     public String accesssdenied()
43     {
44         return "403";
45     }
46 }
47
```

```
RoleRepository.java X JunitService.java JunitServiceImpl.java SpringSecurityProjectApplication.java X
1 package com.hcl;
2
3 import org.springframework.boot.SpringApplication;
4
5
6
7 @SpringBootApplication
8 public class SpringSecurityProjectApplication {
9     public static void main(String[] args) {
10         SpringApplication.run(SpringSecurityProjectApplication.class, args);
11
12         String password = "pass";
13         String password1 = "word";
14         String password2 = "login";
15         BCryptPasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
16         String hashedPassword = passwordEncoder.encode(password);
17         System.out.println("=====");
18         System.out.println(hashedPassword);
19         System.out.println("=====");
20         hashedPassword = passwordEncoder.encode(password1);
21         System.out.println("=====");
22         System.out.println(hashedPassword);
23         System.out.println("=====");
24         hashedPassword = passwordEncoder.encode(password2);
25         System.out.println("=====");
26         System.out.println(hashedPassword);
27         System.out.println("=====");
28     }
29 }
30

application.properties X 403.jsp X
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <title>Error Page</title>
7 </head>
8 <body>
9     <h3>You do not have permission to access this page!</h3>
10    <form action="/Logout" method="post">
11        <input type="submit" value="Sign in as different user" /> <input
12            type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}" />
13    </form>
14 </body>
15 </html>
16
```

```

WebSecurityConfig.java X
1 package com.hcl.config;
2
3 import org.springframework.beans.factory.annotation.Autowired;
12
13 @Configuration
14 public class WebSecurityConfig extends WebSecurityConfigurerAdapter{
15     @Autowired
16     UserDetailsService userDetailsService;
17
18     @Bean(name="passwordEncoder")
19     public PasswordEncoder passwordencoder(){
20         return new BCryptPasswordEncoder();
21     }
22
23     @Override
24     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
25         auth.userDetailsService(userDetailsService).passwordEncoder(passwordencoder());
26     }
27
28     @Override
29     protected void configure(HttpSecurity http) throws Exception {
30         http.authorizeRequests()
31             .antMatchers("/hello").access("hasRole('ROLE_ADMIN')")
32             .antMatchers("/hellouser").access("hasRole('ROLE_USER')")
33             .anyRequest().permitAll()
34             .and()
35             .formLogin().loginPage("/login")
36             .usernameParameter("username").passwordParameter("password")
37             .and()
38             .logout().logoutSuccessUrl("/login?logout")
39             .and()
40             .exceptionHandling().accessDeniedPage("/403")
41             .and()
42             .csrf();
43     }
44 }
45 }
46

```

```

WebSecurityConfig.java JunitController.java JunitEntity.java Role.java JunitRepository.java X
1 package com.hcl.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
6
7 @Repository
8 public interface JunitRepository extends JpaRepository<JunitEntity, Integer> {
9     JunitEntity findByUsername(String username);
10 }
11

```

```
RoleRepository.java  JunitService.java  JunitServiceImpl.java X
1 package com.hcl.service;
2
3 import java.util.HashSet;
4
5 @Service
6 @Transactional
7 public class JunitServiceImpl implements JunitService, UserDetailsService {
8     @Autowired
9     private JunitRepository repository;
10
11     @Override
12     public JunitEntity findIfUserExists(String userid) {
13         return repository.findByUsername(userid);
14     }
15
16     @Override
17     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
18         JunitEntity user = repository.findByUsername(username);
19         Set<GrantedAuthority> grantedAuthorities = new HashSet<>();
20         System.out.println(user.getRoles().toString());
21         for (Role role : user.getRoles())
22         {
23             grantedAuthorities.add(new SimpleGrantedAuthority(role.getName()));
24         }
25         return new org.springframework.security.core.userdetails.User(user.getUsername(), user.getPassword(), grantedAuthorities);
26     }
27 }
28
29 }
```

```
RoleRepository.java  JunitService.java X
1 package com.hcl.service;
2
3 import com.hcl.entity.JunitEntity;
4
5 public interface JunitService {
6     public JunitEntity findIfUserExists(String userid);
7 }
8 }
```