

Implement Spring Security with Authentication Source Code

Developer:Nadikatla Gopi Manikanta

AppErrorController:

```
packagecom.example.SpringSecurityManager.controllers;

importorg.springframework.boot.web.servlet.error.ErrorController;

importorg.springframework.web.bind.annotation.RequestMapping;

public class AppErrorController implements ErrorController {

    @RequestMapping("/error")

    public String handleError() {

        //do something like logging

        return "error";

    }

    @Override

    public String getErrorPath() {

        return null;

    }

}
```

MainController:

```
packagecom.example.SpringSecurityManager.controllers;

importjava.util.ArrayList;

importorg.slf4j.Logger;

importorg.slf4j.LoggerFactory;

importorg.springframework.beans.factory.annotation.Autowired;

importorg.springframework.stereotype.Controller;

importorg.springframework.ui.ModelMap;

importorg.springframework.web.bind.annotation.GetMapping;
```

Implement Spring Security with Authentication Source Code

```
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.SessionAttributes;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;
import com.example.SpringSecurityManager.entities.User;
import com.example.SpringSecurityManager.services.UserService;

@Controller

public class MainController {

    @Autowired

    private UserService userService;

    Logger logger = LoggerFactory.getLogger(MainController.class);

    String currID = null;

    @GetMapping(value="/")

    public String showHomePage(ModelMap model, @RequestParam(value="name", required=false,
    defaultValue="World") String name){

        model.addAttribute("name", name);

        return "home";

    }

    @PostMapping(value="/index")
```

Implement Spring Security with Authentication Source Code

```
        public String showIndexPage(@RequestParam("namellogin") String namellogin,
        @RequestParam("passwordlogin") String passwordlogin, ModelMapmodelMap)

        {

try {

        User u = userService.GetUserByName(namellogin);

        if(u.getName().equals(namellogin) &&u.getPassword().equals(passwordlogin)) {

return "index";

        }

        else

        {

        return "home";

        }

        }

catch(NullPointerException e) {
```

Implement Spring Security with Authentication Source Code

```
        return "home";

    }

}

    public boolean isNumber(String s)
    {

        if(s == null)

            return false;

        try

        {

            double db = Double.parseDouble(s);

        }

        catch(NumberFormatException e)

        {
```

Implement Spring Security with Authentication Source Code

```
return false;
```

```
}
```

```
return true;
```

```
}
```

```
@PostMapping("/update")
```

```
public String saveDetails(@RequestParam("id") String id, ModelMapmodelMap) {
```

```
    try
```

```
    {
```

```
        User user = userService.getUserById(Integer.valueOf(id));
```

```
        ArrayList<User>userList = new ArrayList<>();
```

```
        if(user != null)
```

```
{
```

Implement Spring Security with Authentication Source Code

```
        userList.add(user);
```

```
        Iterable<User> users = userList;
```

```
        currID = id;
```

```
        modelMap.put("user", users);
```

```
    }
```

```
        else
```

```
            return "nouser";
```

```
        }
```

```
    catch (NumberFormatException e)
```

```
    {
```

```
        // TODO Auto-generated catch block
```

```
        return "nouser";
```

```
    }
```

Implement Spring Security with Authentication Source Code

```
catch (Exception e)
```

```
{
```

```
// TODO Auto-generated catch block
```

```
    e.printStackTrace();
```

```
}
```

```
modelMap.put("ID", id);
```

```
return "update";
```

```
}
```

```
@PostMapping("/update2")
```

```
public String updateDetails(@RequestParam("nameedit") String nameedit,  
    @RequestParam("emailedit") String emailedit, @RequestParam("passwordedit") String passwordedit,  
    ModelMapmodelMap) {
```

Implement Spring Security with Authentication Source Code

```
ArrayList<User> userList = new ArrayList<>();
```

```
try
```

```
{
```

```
User u = userService.getUserById(Integer.valueOf(currID));
```

```
userService.setUser(u, nameedit, emailedit, passwordedit);
```

```
userList.add(u);
```

```
Iterable<User> users = userList;
```

```
modelMap.put("user", users);
```

```
}
```

```
catch (NumberFormatException e)
```

```
{
```

```
e.printStackTrace();
```

```
}
```


Implement Spring Security with Authentication Source Code

```
        catch(Exception e)

{

        e.printStackTrace();

}

        modelMap.put("IDedit", currID);


        return "update2";

}

}
```

User Controller:

```
package com.example.SpringSecurityManager.controllers;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.ModelMap;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.ResponseBody;

import com.example.SpringSecurityManager.entities.User;

import com.example.SpringSecurityManager.services.UserService;
```

Implement Spring Security with Authentication Source Code

```
import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@Controller

public class UserController {

    @Autowired

    private UserService userService;

    Logger logger = LoggerFactory.getLogger(UserController.class);

    @GetMapping("/users")

    public String showUsers(ModelMap model) {

        logger.info("Getting all Users");

        Iterable<User> users = userService.GetAllUsers();

        logger.info("Passing users to view");

        model.addAttribute("users", users);

        return "users";

    }

}
```

UserExceptionHandler:

```
package com.example.SpringSecurityManager.controllers;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.ControllerAdvice;

import org.springframework.web.bind.annotation.ExceptionHandler;

import com.example.SpringSecurityManager.exceptions.UserNotFoundException;

@ControllerAdvice

public class UserExceptionHandler {
```

Implement Spring Security with Authentication Source Code

```
@ExceptionHandler(value = UserNotFoundException.class)

public ResponseEntity<Object> exception(UserNotFoundException exception) {

return new ResponseEntity<>("User not found", HttpStatus.NOT_FOUND);

}

}
```

User Entity:

```
package com.example.SpringSecurityManager.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity // This tells Hibernate to make a table out of this class

public class User {

@Id

@GeneratedValue(strategy=GenerationType.AUTO)


private Integer id;


private String name;


private String email;
```

Implement Spring Security with Authentication Source Code

```
private String password;
```

```
public String getPassword() {
```

```
    return password;
```

```
}
```

```
public void setPassword(String password) {
```

```
    this.password = password;
```

```
}
```

```
public Integer getId() {
```

```
    return id;
```

```
}
```

Implement Spring Security with Authentication Source Code

```
public void setId(Integer id) {
```

```
    this.id = id;
```

```
}
```

```
public String getName() {
```

```
    return name;
```

```
}
```

```
public void setName(String name) {
```

```
    this.name = name;
```

```
}
```

```
public String getEmail() {
```

```
    return email;
```

```
}
```

```
public void setEmail(String email) {
```

Implement Spring Security with Authentication Source Code

```
this.email = email;
```

```
}
```

```
@Override
```

```
public String toString() {
```

```
    return (id.toString() + " " + name + " " + email + " " + password);
```

```
}
```

```
}
```

UserNotFoundException:

```
packagecom.example.SpringSecurityManager.exceptions;
```

```
public class UserNotFoundException extends RuntimeException {
```

```
    private static final long serialVersionUID = 1L;
```

```
}
```

UserRepository:

```
packagecom.example.SpringSecurityManager.repositories;
```

```
importorg.springframework.data.repository.CrudRepository;
```

```
importcom.example.SpringSecurityManager.entities.User;
```

```
public interface UserRepository extends CrudRepository<User, Integer> {
```

```
    public User findByName(String name);
```

```
}
```

Implement Spring Security with Authentication Source Code

UserService:

```
package com.example.SpringSecurityManager.services;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.SpringSecurityManager.entities.User;
import com.example.SpringSecurityManager.repositories.UserRepository;

@Service

public class UserService {

    @Autowired

    private UserRepository userRepository;

    public Iterable<User> GetAllUsers()

    {

        return userRepository.findAll();

    }


    public User GetUserByName(String name) {

        User foundUser = userRepository.findByName(name);

        return foundUser;

    }


    public User GetUserById(int id) throws Exception {

        Optional<User> foundUser = userRepository.findById(id);
```

Implement Spring Security with Authentication Source Code

//TODO: we need to decide how to handle a "Not Found" condition

```
if(!foundUser.isPresent())
```

```
    return null;
```

```
    return(foundUser.get());
```

```
}
```

```
public void updateUser(User usertoUpdate) {
```

```
    userRepository.save(usertoUpdate);
```

```
}
```

```
public void setUser(User u, String name, String email, String password) {
```

```
    //u.setId(id);
```

```
    u.setName(name);
```

```
    u.setEmail(email);
```

```
    u.setPassword(password);
```

```
    updateUser(u);
```

```
}
```

```
}
```


Implement Spring Security with Authentication Source Code

UserManagerApplication:

```
package com.example.SpringSecurityManager;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class UserManagerApplication {

    public static void main(String[] args) {

        SpringApplication.run(UserManagerApplication.class, args);

    }

}
```

Home.jsp:

```
<html>

<head>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <style>

        .center {

            text-align: center;

        }

    </style>

</head>

<body style="background-color:lightblue;">

    <div class="d-flex justify-content-center">
```

Implement Spring Security with Authentication Source Code

```
<div class="w-75 p-3">

<div class="center">

    <h1 class="display-4">Spring Security</h1>

    <div class="jumbotron">

        <p class="lead">Login below to access the user's table</p>

        <form method="post" action="index">

            <input type="text" id="namellogin" name="namellogin"
placeholder="Name" required>

            <input type="text" id="passwordlogin"
name="passwordlogin" placeholder="Password" required>

            <input type="submit" value="Enter" class="btn btn-
primary mb-2" />

        </form>

    </div>

</div>

</div>

</div>

</div>

</body>

</html>
```

Index.Jsp:

```
<html>

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

Implement Spring Security with Authentication Source Code

```
<style>
```

```
.center {
```

```
    text-align: center;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body style="background-color:lightblue;">
```

```
<div class="d-flex justify-content-center">
```

```
    <div class="w-75 p-3">
```

```
        <div class="center">
```

```
            <h1 class="display-4">Search for a User By ID</h1>
```

```
        <div class="jumbotron">
```

Implement Spring Security with Authentication Source Code

```
<h2 class="hello-title">Login Success</h2>
```

```
<p class="lead">View user table <a href="/users">here</a></p>
```

```
<br><br>
```

```
<form method="post" action="update">
```

```
<p class="lead">Enter an id from the table: <p>
```

```
<input type="text" id="id" name="id" placeholder="Type here" required>
```

```
<input type="submit" value="Enter" class="btn btn-primary mb-2"/>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body></html>
```

Implement Spring Security with Authentication Source Code

Update1.Jsp:

```
<%@ tagliburi="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<html>
```

```
<head>
```

```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
    <style>
```

```
    table, th, td {
```

```
        border: 1px solid black;
```

```
        margin: auto;
```

```
    }
```

```
    .center {
```

```
        text-align: center;
```

```
    }
```

```
</style>
```

```
</head>
```

```
<body style="background-color:lightblue;">
```

```
    <div class="d-flex justify-content-center">
```

```
        <div class="w-75 p-3">
```

```
            <div class="center">
```

```
                <div class="jumbotron">
```

```
                    <h2 class="display-4">Update Table</h2>
```

Implement Spring Security with Authentication Source Code

```
<p class="lead"> User ID: ${ID}</p>

<table style="float:inherit">

<tr><th>ID</th><th>Name</th><th>Email</th><th>Password</th></tr>

<c:forEach items="${user}" var="userE" varStatus="count">

<tr id="${count.index}">

<td>${userE.id}</td>

<td>${userE.name}</td>

<td>${userE.email}</td>

<td>${userE.password}</td>

</tr>

</c:forEach>

</table>

<br><br>

<form method="post" action="update2">

<br><h3>Edit user: ${ID}</h3>

<input type="text" id="nameedit" name="nameedit" placeholder="Name" required>

<input type="text" id="emailedit" name="emailedit" placeholder="Email" required>

<input type="text" id="passwordedit" name="passwordedit" placeholder="Password" required>

<input type="submit" value="Enter" class="btn btn-primary mb-2"/>

</form>

</div>

</div>

</div>

</div>

</body></html>
```

Implement Spring Security with Authentication Source Code

Update2.Jsp:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<html>
```

```
<head>
```

```
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
    <style>
```

```
    table, th, td {
```

```
        border: 1px solid black;
```

```
        margin: auto;
```

```
    }
```

```
    .center {
```

```
        text-align: center;
```

```
    }
```

```
    </style>
```

```
</head>
```

```
<body style="background-color:lightblue;">
```

```
    <div class="d-flex justify-content-center">
```

```
        <div class="w-75 p-3">
```

```
            <div class="center">
```

```
                <h2 class="display-4">Successfully Updated User</h2>
```

```
            <div class="jumbotron">
```

```
                <p class="lead"> User ID: ${IDedit}</p>
```

Implement Spring Security with Authentication Source Code

```
<div>

    <table style="float:inherit">

        <tr><th>ID</th><th>Name</th><th>Email</th><th>Password</th></tr>

        <c:forEach items="{user}" var="userE" varStatus="count">

            <tr id="{count.index}">

                <td>${userE.id}</td>

                <td>${userE.name}</td>

                <td>${userE.email}</td>

                <td>${userE.password}</td>

            </tr>

        </c:forEach>

    </table>

</div>

<br><br>

<h3>Return to Homepage</h3>

<div>

    <a href="/">Return</a>

</div>

</div>

</div>

</div>

</div>

</body>

</html>
```


Implement Spring Security with Authentication Source Code

User.Jsp:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
```

```
<style>
```

```
table, th, td {
```

```
border: 1px solid black;
```

```
margin: auto;
```

```
}
```

```
.center {
```

```
text-align: center;
```

```
}
```

```
</style>
```

Implement Spring Security with Authentication Source Code

```
</head>
```

```
<body style="background-color:lightblue;">
```

```
<div class="d-flex justify-content-center">
```

```
<div class="w-75 p-3">
```

```
<div class="center">
```

```
<div class="jumbotron">
```

```
<h2 class="display-4">Users</h2>
```

```
<table style="float:inherit">
```

```
<tr><th>ID</th><th>Name</th><th>Email</th><th>Password</th></tr>
```

```
<c:forEach items="${users}" var="user" varStatus="count">
```

```
<tr id="${count.index}">
```

```
<td>${user.id}</td>
```

```
<td>${user.name}</td>
```

Implement Spring Security with Authentication Source Code

```
<td>${user.email}</td>
```

```
<td>${user.password}</td>
```

```
</tr>
```

```
</c:forEach>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

UserManagerApplicationTests:

```
package com.example.UserManager;
```

```
import org.junit.jupiter.api.Test;
```

```
import org.springframework.boot.test.context.SpringBootTest;
```

Implement Spring Security with Authentication Source Code

```
@SpringBootTest
```

```
class
```

```
UserManagerApplicationTests {
```

```
@Test
```

```
void contextLoads() {
```

```
}
```

```
}
```

Application Property:

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/phase2
```

```
spring.datasource.username=root
```

```
spring.datasource.password=root
```

```
logging.level.org.springframework.web: DEBUG
```

```
spring.mvc.view.prefix=/WEB-INF/jsp/
```

```
spring.mvc.view.suffix=.jsp
```

```
server.port=8090
```