#### Developer:Nadikatla Gopi Manikanta

#### **AppErrorController:**

```
packagecom.example.SpringSecurityManager.controllers;
importorg. spring framework. boot. we b. servlet. error. Error Controller;\\
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;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
importorg.springframework.beans.factory.annotation.Autowired;
importorg.springframework.stereotype.Controller;
importorg.springframework.ui.ModelMap;
importorg.springframework.web.bind.annotation.GetMapping;
```

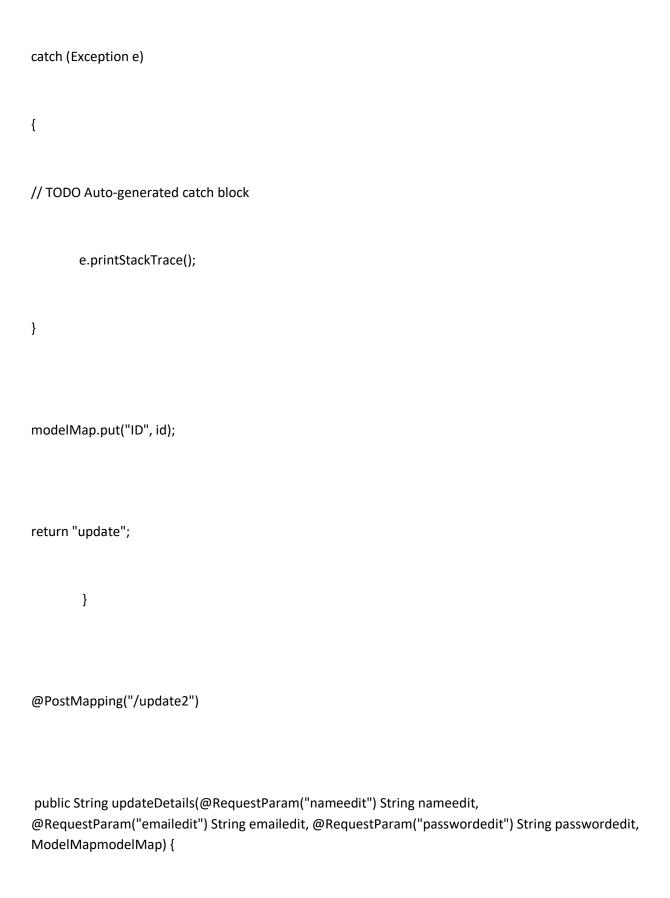
```
importorg.springframework.web.bind.annotation.PathVariable;
importorg.springframework.web.bind.annotation.PostMapping;
importorg.springframework.web.bind.annotation.RequestMapping;
importorg.springframework.web.bind.annotation.RequestMethod;
importorg.springframework.web.bind.annotation.RequestParam;
importorg.springframework.web.bind.annotation.ResponseBody;
importorg.springframework.web.bind.annotation.SessionAttributes;
importorg.springframework.web.servlet.mvc.support.RedirectAttributes;
importcom.example.SpringSecurityManager.entities.User;
importcom.example.SpringSecurityManager.services.UserService;
@Controller
public class MainController {
       @Autowired
       privateUserServiceuserService;
        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")
```

```
public String showIndexPage(@RequestParam("namelogin") String namelogin,
@RequestParam("passwordlogin") String passwordlogin, ModelMapmodelMap)
        {
try {
User u = userService.GetUserByName(namelogin);
       if(u.getName().equals(namelogin) &&u.getPassword().equals(passwordlogin)) {
return "index";
}
else
{
       return "home";
        }
}
catch(NullPointerException e) {
```

```
return "home";
}
}
       publicbooleanisNumber(String s)
       {
       if(s == null)
return false;
try
{
doubledb = Double.parseDouble(s);
}
catch(NumberFormatException e)
{
```

```
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)
{
```

```
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";
}
```



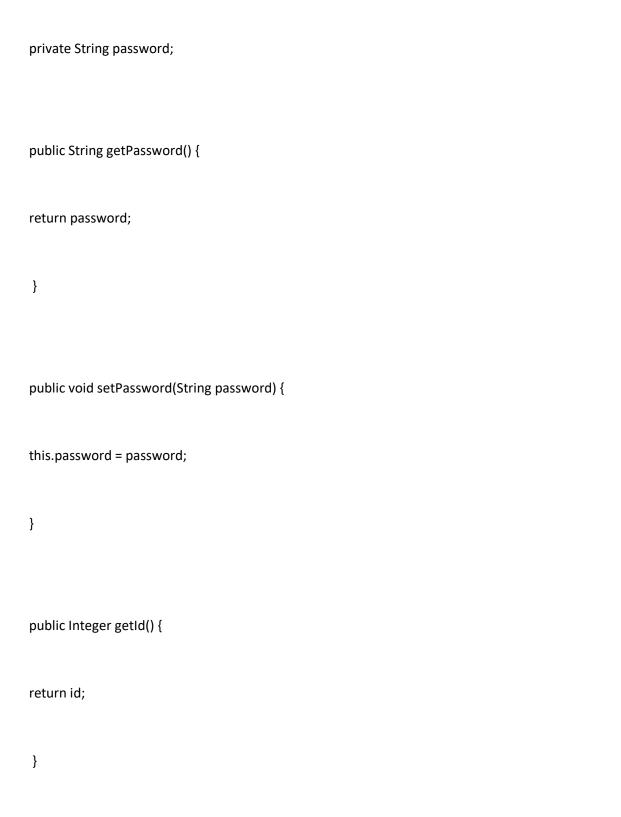
```
ArrayList<User>userList = new ArrayList<>();
try
                {
User u = userService.GetUserById(Integer.valueOf(currID));
               userService.setUser(u, nameedit, emailedit, passwordedit);
               userList.add(u);
Iterable<User> users = userList;
modelMap.put("user", users);
}
catch (NumberFormatException e)
                {
       e.printStackTrace();
}
```

```
catch(Exception e)
{
                      e.printStackTrace();
        }
               modelMap.put("IDedit", currID);
               return "update2";
}
}
User Controller:
packagecom.example.SpringSecurityManager.controllers;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
importorg.springframework.beans.factory.annotation.Autowired;
importorg.springframework.stereotype.Controller;
importorg.springframework.ui.ModelMap;
importorg.springframework.web.bind.annotation.GetMapping;
importorg.springframework.web.bind.annotation.PathVariable;
importorg.springframework.web.bind.annotation.ResponseBody;
importcom.example.SpringSecurityManager.entities.User;
importcom.example.SpringSecurityManager.services.UserService;
```

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
@Controller
public class UserController {
       @Autowired
       privateUserServiceuserService;
        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";
}
       }
UserExceptionController:
packagecom.example.SpringSecurityManager.controllers;
importorg.springframework.http.HttpStatus;
importorg.springframework.http.ResponseEntity;
importorg.springframework.web.bind.annotation.ControllerAdvice;
importorg.springframework.web.bind.annotation.ExceptionHandler;
import com.example.SpringSecurityManager.exceptions.UserNotFoundException;
@ControllerAdvice
```

public class UserExceptionController {

```
@ExceptionHandler(value = UserNotFoundException.class)
publicResponseEntity<Object> exception(UserNotFoundException exception) {
return new ResponseEntity<>("User not found", HttpStatus.NOT_FOUND);
  }
}
User Entitiy:
packagecom.example.SpringSecurityManager.entities;
importjavax.persistence.Entity;
importjavax.persistence.GeneratedValue;
importjavax.persistence.GenerationType;
importjavax.persistence.ld;
@Entity // This tells Hibernate to make a table out of this class
public class User {
@ld
@GeneratedValue(strategy=GenerationType.AUTO)
private Integer id;
private String name;
private String email;
```



```
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) {
```

```
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:
package com. example. Spring Security Manager. repositories;\\
importorg.springframework.data.repository.CrudRepository;
importcom.example.SpringSecurityManager.entities.User;
public interface UserRepository extends CrudRepository<User, Integer> {
public User findByName(String name);
}
```

#### **UserService:**

```
packagecom.example.SpringSecurityManager.services;
importjava.util.Optional;
importorg.springframework.beans.factory.annotation.Autowired;
importorg.springframework.stereotype.Service;
importcom.example.SpringSecurityManager.entities.User;
import com. example. Spring Security Manager. repositories. User Repository;\\
@Service
public class UserService {
@Autowired
privateUserRepositoryuserRepository;
publicIterable<User>GetAllUsers()
  {
returnuserRepository.findAll();
}
public User GetUserByName(String name) {
User foundUser = userRepository.findByName(name);
returnfoundUser;
}
public User GetUserById(int id) throws Exception {
Optional<User>foundUser = userRepository.findById(id);
```

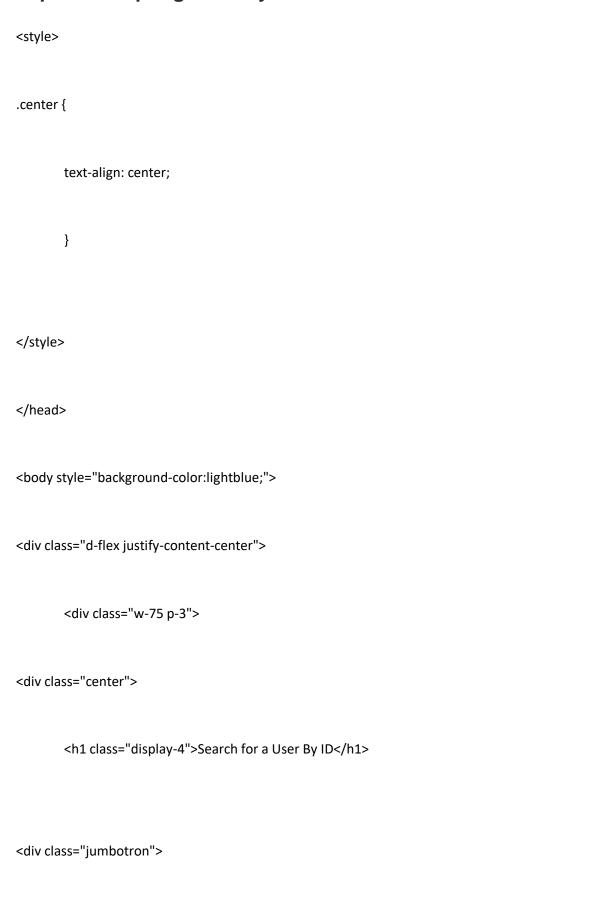
```
//TODO: we need to decide how to handle a "Not Found" condition
if(!foundUser.isPresent())
               return null;
       return(foundUser.get());
}
public void UpdateUser(User usertoUpdate) {
userRepository.save(usertoUpdate);
}
public void setUser(User u, String name, String email, String password) {
       //u.setId(id);
u.setName(name);
u.setEmail(email);
       u.setPassword(password);
UpdateUser(u);
}
}
```

#### **UserManagerApplication:**

```
packagecom.example.SpringSecurityManager;
importorg.springframework.boot.SpringApplication;
importorg.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"</pre>
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">
```

```
<div class="w-75 p-3">
              <div class="center">
                     <h1 class="display-4">Spring Security</h1>
                            <div class="jumbotron">
                                    Login below to access the user's table
                                    <form method="post" action="index">
                                           <input type="text" id="namelogin" name="namelogin"
placeholder="Name" required>
                                           <input type="text" id="passwordlogin"
name="passwordlogin" placeholder="Password" required>
                                           <input type="submit" value="Enter" class="btnbtn-
primary mb-2"/>
                                    </form>
                            </div>
              </div>
              </div>
       </div>
</body>
</html>
Index.Jsp:
<html>
<head>
k rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
```

crossorigin="anonymous">



<h2 class="hello-title">Login Success</h2>
View user table <a href="/users">here</a>
> >
<form action="update" method="post"></form>
Enter an id from the table:
<input id="id" name="id" placeholder="Type here" required="" type="text"/>
<input class="btnbtn-primary mb-2" type="submit" value="Enter"/>

#### **Update1.Jsp:**

```
<%@ tagliburi="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>
       <link rel="stylesheet"</pre>
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>
```

```
 User ID: ${ID}
                       IDNameEmailPassword
                       <c:forEach items="${user}" var="userE" varStatus="count">
                       ${userE.id}
                       ${userE.name}
                       ${userE.email}
                       ${userE.password}
                       </c:forEach>
                       <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="emaileditplaceholder="Email" required>
      <input type="text" id="passwordedit" name="passwordedit" placeholder="Password"required>
           <input type="submit" value="Enter" class="btnbtn-primary mb-2"/>
                       </form>
                 </div>
           </div>
           </div>
     </div>
</body></html>
```

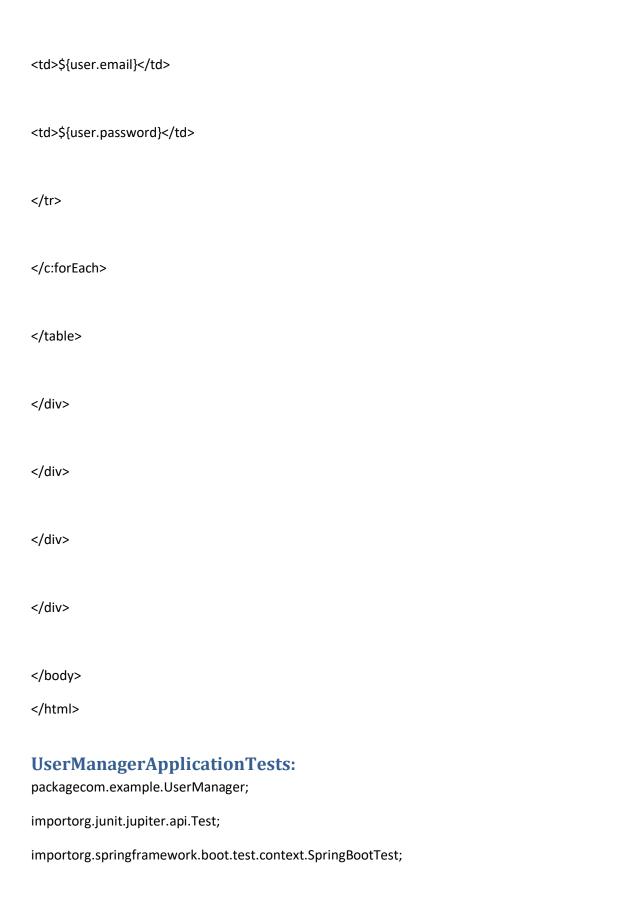
#### **Update2.Jsp:**

```
<%@ tagliburi="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html>
<head>
       <link rel="stylesheet"</pre>
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">
                       User ID: ${IDedit}
```

```
<div>
                   IDNameEmailPassword
                   <c:forEach items="${user}" var="userE" varStatus="count">
                   ${userE.id}
                   ${userE.name}
                   ${userE.email}
                   ${userE.password}
                   </c:forEach>
                   </div>
              <br><br>>
              <h3>Return to Homepage</h3>
              <div>
                   <a href="/">Return</a>
              </div>
         </div>
    </div>
    </div>
    </div>
</body>
</html>
```

User.Jsp:
<%@ tagliburi="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html></html>
<head></head>
< link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4lWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
<style></td></tr><tr><td>table, th, td {</td></tr><tr><td></td></tr><tr><td>border: 1px solid black;</td></tr><tr><td></td></tr><tr><td>margin: auto;</td></tr><tr><td></td></tr><tr><td></td></tr><tr><td>}</td></tr><tr><td></td></tr><tr><td>.center {</td></tr><tr><td></td></tr><tr><td>text-align: center;</td></tr><tr><td></td></tr><tr><td>}</td></tr><tr><td>a latulas</td></tr><tr><td></style>





```
@SpringBootTest

class

UserManagerApplicationTests {
    @Test

voidcontextLoads() {
    }
}

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

spring.mvc.view.suffix=.jsp

server.port=8090

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

logging.level.org.springframework.web: DEBUG