

**Searching for a specific
user and updating the
user information project
source code**

Prepared By: G.PADMAJA YADAV

MainController.java

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

```
import java.util.ArrayList;
```

```
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.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 com.example.UserManager.entities.User; import  
com.example.UserManager.services.UserService;  
@Controller public class MainController {
```

```
@Autowired
```

```
private UserService userService;
```

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

```
String currID = null;
```

```
@GetMapping(value="/")
```

```
public String showIndexPage(ModelMap model,
```

```
    @RequestParam(value="name", required=false,  
defaultValue="World") String name){        model.addAttribute("name",  
name);
```

```
    return "index";
```

```
}
```

```
public boolean isNumber(String s)
{
    if(s == null)    return false; try
    {
        double db = Double.parseDouble(s);
    }

    catch(NumberFormatException e)
    {
        return false;
    }

    return true;
}
```

```
}
```

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

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

```
    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";
}

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, ModelMap modelMap) {
```

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

```
        try
```

```
        {    User u
```


=

```
userService.GetUser
```

```
serById(Integer.v
```

```
alueOf(currID));
```

```
userService.setUs
```

```
er(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";
```

```
}  
  
}
```

UserController.java

```
package com.example.UserManager.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.UserManager.entities.User; import
com.example.UserManager.services.UserService;
```

```
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.java

```
package com.example.UserManager.controllers; import
org.springframework.http.HttpStatus;           import
org.springframework.http.ResponseEntity;       import
org.springframework.web.bind.annotation.ControllerAdvi
ce;                                           import
org.springframework.web.bind.annotation.ExceptionHand
ler;

import com.example.UserManager.exceptions.UserNotFoundException;

@ControllerAdvice public class
UserExceptionHandler {
```

```
    @ExceptionHandler(value = UserNotFoundException.class)    public  
    ResponseEntity<Object> exception(UserNotFoundException exception) {  
    return new ResponseEntity<>("User not found", HttpStatus.NOT_FOUND);  
    }  
}
```

User.java

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

```
    private String password;
```

```
    public String getPassword() {  
return password;  
    }
```

```
    public void setPassword(String password) {  
this.password = password;  
    }
```

```
    public Integer getId() {  
return id;  
    }
```



```
}
```

```
    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.java package

```
com.example.UserManager.exceptions;
```

```
public class UserNotFoundException extends RuntimeException {  
    private static final long serialVersionUID = 1L;  
}
```

- ***UserRepository.java*** package

```
com.example.UserManager.repositories;
```

```
import org.springframework.data.repository.CrudRepository;
```

```
import com.example.UserManager.entities.User;
```

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

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

```
}
```

```
UserService.java package com.example.UserManager.services;
```

```
import java.util.Optional;
```

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

```
import com.example.UserManager.entities.User; import  
com.example.UserManager.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);
```

```
        //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.java package

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

```
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);  
}  
  
}
```

Application.properties

```
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
```

Index.jsp

```
<html>
```

```
<head>
```

```
    <style>
```

```
        .center {    text-align: center;
```

```
        }
```

```
    </style>
```

```
</head>
```

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

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

```
<h1>User Manager</h1>
```

```
<h2 class="hello-title">Search for a User and Update  
Information</h2>
```

```
<a href="/users">View user table</a>
```

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

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

```
Enter an id from the user table: <input type="text" id="id" name="id"  
placeholder="Type here" required><input type="submit" value="Enter" />
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

Users.jsp

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

```
<head> <style> table,
```

```
th, td { border: 1px
```

```
solid black;
```

```
margin: auto;
```

```
}
```

```
.center { text-align: center;
```

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

</tr>

</c:forEach>

</table>

</div>

</body>

</html>

UserManagerApplicationTest.java

```
package com.example.UserManager; import
org.junit.jupiter.api.Test; import
org.springframework.boot.test.context.SpringBootTest;
@SpringBootTest class

userManagerApplicationTests {

    @Test

    void contextLoads() {

    }

}
```

=====X=====