**Handling User Authentication**

**Source Code**

**1. AuthenticationApplication**

package com.example.Authentication;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Import;

import com.example.Authentication.controllers.LoginController;

import com.example.Authentication.controllers.UserDNEController;

import com.example.Authentication.entities.User;

import com.example.Authentication.exceptions.UserNotFoundException;

import com.example.Authentication.services.UserService;

@SpringBootApplication

@Import({

UserNotFoundException.class,

UserService.class,

LoginController.class,

User.class,

UserDNEController.class

})

public class AuthenticationApplication {

public static void main(String[] args) {

SpringApplication.run(AuthenticationApplication.class, args);

System.out.println("Running...");

}

}

**2. LoginController**

package com.example.Authentication.controllers;

import javax.servlet.http.HttpServletRequest;

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.PostMapping;

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

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

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

import com.example.Authentication.entities.User;

import com.example.Authentication.exceptions.UserNotFoundException;

import com.example.Authentication.services.UserService;

@Controller

public class LoginController {

@Autowired

UserService userService;

Logger log = LoggerFactory.getLogger(LoginController.class);

@GetMapping("/index")

public String showGreeting(ModelMap map) {

return "index";

}

@GetMapping("/loginform")

public String showLogin(ModelMap map) {

return "loginform";

}

@PostMapping("/loginform")

public String submitLogin(ModelMap map, @RequestParam String username, @RequestParam String password){

User checkUser;

try {

checkUser= userService.GetUserByName(username);

if (userService.verifyPassword(username, password) == false) {

log.info(username + " - " + password);

map.addAttribute("error", "Error Wrong password");

return "deniedAccess";

}

else

return "success";

} catch (UserNotFoundException unfe) {

throw unfe;

}

}

}

**3. UserDNEController**

package com.example.Authentication.controllers;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.ui.ModelMap;

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

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

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

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

import com.example.Authentication.entities.User;

import com.example.Authentication.exceptions.UserNotFoundException;

import com.example.Authentication.services.UserService;

@ControllerAdvice

public class UserDNEController {

Logger log = LoggerFactory.getLogger(LoginController.class);

@ExceptionHandler(value = UserNotFoundException.class)

public String errorLogin(UserNotFoundException dne){

log.info("error found");

return "deniedAccess";

}

}

**4.User.java**

package com.example.Authentication.entities;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

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

@Table(name = "users")

public class User {

@Id

@GeneratedValue(strategy=GenerationType.IDENTITY)

private Integer id;

private String name;

private String email;

private String password;

public User(String name, String email, String password) {

this.name = name;

this.email = email;

this.password = password;

}

public User() {

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

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);

}

}

**5. UserNotFoundException**

**package** com.example.Authentication.exceptions;

**public** **class** UserNotFoundException **extends** RuntimeException {

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

}

**6. UserRepository**

package com.example.Authentication.repositories;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;

import com.example.Authentication.entities.User;

public interface UserRepository extends CrudRepository<User, Integer> {

public Optional<User> findByName(String name);

}

**7. UserService**

package com.example.Authentication.services;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.example.Authentication.entities.User;

import com.example.Authentication.exceptions.UserNotFoundException;

import com.example.Authentication.repositories.UserRepository;

@Service

public class UserService {

@Autowired

private UserRepository userRepository;

public Iterable<User> GetAllUsers()

{

return userRepository.findAll();

}

public User GetUserByName(String name) {

Optional<User> foundUser = userRepository.findByName(name);

if (!foundUser.isPresent()) {

throw new UserNotFoundException();

}

return(foundUser.get());

}

public boolean verifyPassword(String username, String password) {

boolean verified = false;

User user = GetUserByName(username);

if (user.getPassword().equals(password)) {

verified = true;

}

return verified;

}

}

**8.application.properties**

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/phase2

spring.datasource.username=root

spring.datasource.password=Root@1999

logging.level.org.springframework.web: DEBUG

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

spring.mvc.view.suffix=.jsp

server.port=8080

server.error.whitelabel.enabled=false

**webapp**

**1.index.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Home Page</title>

</head>

<body style="text-align: *center*">

<h2 >User Authentication</h2>

<a href=*"loginform"*>Login</a>

</body>

</html>

**2.loginform.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>User Login</title>

</head>

<body style="text-align: *center*">

<h2 >Login Page</h2>

<div style="color:*red*;">

${error}

</div>

<form action=*"loginform"* method=*'post'*>

<label for=*"username"*>Name:</label><br> <input type=*"text"*

id=*"username"* placeholder=*"Name Required"* name=*"username"* required><br>

<br> <label for=*"password"*>Password:</label><br> <input

type=*"text"* id=*"password"* placeholder=*"Password Required"*

name=*"password"* required><br>

<br> <input type=*"submit"* value=*"Submit"*>

</form>

</body>

</html>

**3.success.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Login Successful</title>

</head>

<body style="text-align: *center*">

<form style="text-align: *center*; margin-left: *auto*; margin-right: *auto*"

action=*"loginform"* method=*"get"*>

<label for=*"username"*></label> <input type=*"hidden"* name=*"username"*>

<br> <br> <label for=*"password"*></label> <input

type=*"hidden"* name=*"password"*> <br> <br>

</form>

<h2 style="text-align: *center*">Login Successful</h2>

<h2 style="text-align: *center*">Login Once again</h2>

<a href=*"loginform"*>Login</a>

<br>

</body>

</html>

**4.deniedAcess.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Login Denied</title>

</head>

<body style="text-align: *center*">

<h2>Denied Page</h2>

<p> The username and/or password was incorrect.</p>

<a href=*"loginform"*>Return to Log In</a>

</body>

</html>