

# Handling User Authentication

## Source Code:

### AuthenticationApplication.java

```
package com.project.Authentication;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Import;

import com.project.Authentication.controllers.AuthenticationController;
import com.project.Authentication.entities.User;
import com.project.Authentication.exceptions.UserNotFoundException;
import com.project.Authentication.services.AuthenticationService;

@SpringBootApplication
@Import({
    AuthenticationController.class,
    UserNotFoundException.class,
    AuthenticationService.class,
    User.class
})
public class AuthenticationApplication {

    public static void main(String[] args) {
        SpringApplication.run(AuthenticationApplication.class, args);
    }

}
```

### AuthenticationController.java

```
package com.project.Authentication.controllers;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
```

```

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;

import com.project.Authentication.entities.User;
import com.project.Authentication.services.AuthenticationService;

@Controller
public class AuthenticationController {

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

    @Autowired
    AuthenticationService authService;

    @GetMapping("/")
    public String showGreeting() {
        return "greeting";
    }

    @GetMapping("/Auth")
    public String showLogin() {
        return "authenticate";
    }

    @PostMapping("/Auth")
    public String authenticateUser(@RequestParam("username") String username,
    @RequestParam("password") String pswd) {
        User user = authService.getUserByName(username);
        logger.info(user.getName() + " attempted to login with " + user.getPassword());
        String path = (authService.isValidPassword(pswd, user.getPassword())) ?
"success" : "failure";
        logger.info("The path return: " + path);
        return path;
    }
}

```

## **User.java**

```

package com.project.Authentication.entities;

import javax.persistence.Column;

```

```
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
```

```
@Entity
```

```
@Table(name = "user")
```

```
public class User {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.AUTO)
```

```
    @NotNull
```

```
    private Integer id;
```

```
    @Column(name = "name")
```

```
    @NotNull
```

```
    private String name;
```

```
    @Column(name = "email")
```

```
    @NotNull
```

```
    private String email;
```

```
    @Column(name = "password")
```

```
    @NotNull
```

```
    private String password;
```

```
    public User() {
```

```
        super();
```

```
    }
```

```
    public User(@NotNull String name, @NotNull String password) {
```

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

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

```
    }
```

```
    public User(@NotNull String name, @NotNull String email, @NotNull String password) {
```

```
        super();
```

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

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

```
        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;
    }

    public String getPassword() {
        return password;
    }

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

    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + ", email=" + email + ", password=" +
password + "]\n";
    }
}

```

**UserNotFoundException.java**

```
package com.project.Authentication.exceptions;

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

### **AuthenticatioRepository.java**

```
package com.project.Authentication.repositories;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.project.Authentication.entities.User;

@Repository
public interface AuthenticationRepository extends CrudRepository<User, Integer> {

    public Optional<User> findUserByName(String name);
}
```

### **AuthenticationService.java**

```
package com.project.Authentication.services;

import java.util.Optional;

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

import com.project.Authentication.entities.User;
import com.project.Authentication.exceptions.UserNotFoundException;
import com.project.Authentication.repositories.AuthenticationRepository;

@Service
```

```

public class AuthenticationService {

    @Autowired
    AuthenticationRepository authRepo;

    public User GetUserByName(String name) {
        Optional<User> found = authRepo.findUserByName(name);
        if(found.isPresent()) return found.get();
        else throw new UserNotFoundException();
    }

    public Boolean isValidPassword(String cmp, String actual) {
        return ((cmp.equals(actual)) ? true : false);
    }

}

```

### **applicationProperties.txt**

```

spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/authentication
spring.datasource.username=root
spring.datasource.password=ramani44

```

```

logging.level.org.springframework.web: DEBUG
spring.mvc.view.prefix=/WEB-INF/jsp/
spring.mvc.view.suffix=.jsp
server.port=8090

```

### **authenticate.jsp**

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Authentication Page</title>
</head>
<h2>Login Page</h2>

```

```
<body>
welcome to the authentication page

<form:form action="Auth" method="post" commandName="login">
    <label for="username"> Username:</label>
    <input name="username" id="username" type="text" placeholder="Username"
required/>
    <label for="password">Password:</label>
    <input name="password" id="password" type="password" placholder="Password"
required/>
    <input type="submit" name="Submit"/>
</form:form>
</body>
</html>
```

### **failure.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 Failed</title>
</head>
<body>
<h1>Oops!!!! You Failed in login, Please try again....
</h1><br/>
<a href="/Auth">click here to login again</a>
</body>
</html>
```

### **Greeting.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>Welcome User</title>
</head>
```

```
<h2>Welcome User</h2>
<body>
Welcome You User on the welcome page
<a href="Auth">Login</a>
</body>
</html>
```

### 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>Successful Login Page</title>
</head>
<body>
<h1>Successful Login</h1>
</body>
</html>
```

### POM.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.4.3</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.project</groupId>
    <artifactId>Authentication</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>Authentication</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>1.8</java.version>
    </properties>
```



```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jersey</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
  </dependency>

  <dependency>
    <groupId>javax.xml.bind</groupId>
    <artifactId>jaxb-api</artifactId>
  </dependency>

  <dependency>
    <groupId>org.javassist</groupId>
```

```

        <artifactId>javassist</artifactId>
        <version>3.25.0-GA</version>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok-maven-plugin</artifactId>
        <version>1.18.18.0</version>
        <type>maven-plugin</type>
    </dependency>

    <dependency>
        <groupId>com.h2database</groupId>
        <artifactId>h2</artifactId>
        <scope>runtime</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok-maven-plugin</artifactId>
            <version>1.18.18.0</version>
        </plugin>
    </plugins>
</build>

</project>

```