# SOURCE CODE FOR IMPLEMENT SPRING SECURITY WITH AUTHENTICATION

////////////////////////////////////////////////////////////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
https://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>3.1.0</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>com.example</groupId>
<artifactId>SpringSecurity</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>SpringSecurity</name>
<description>Demo project for Spring Boot</description>
<properties>
<java.version>17</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.security</groupId>
<artifactId>spring-security-test</artifactId>
<scope>test</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
```

//////////////////////////////////////////////////CustomAuthenticationProvider.java :

```
package com.example;
import org.springframework.security.authentication.*;
import org.springframework.security.core.*;
import
org.springframework.security.core.authority.SimpleGrantedAuthority;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;
public class CustomAuthenticationProvider implements
AuthenticationProvider {
List<User> dummyUsers = new ArrayList<>();
public CustomAuthenticationProvider() {
dummyUsers.add(new User("john", "secret", "ROLE_USER"));
dummyUsers.add(new User("admin", "supersecret", "ROLE_ADMIN"));
}
@Override
public Authentication authenticate(Authentication authentication)
throws AuthenticationException {
String name = authentication.getName();
String password = authentication.getCredentials().toString();
//jdk 8 -- stream
Optional<User> authenticatedUser = dummyUsers.stream().filter(
user -> user.getName().equals(name) &&
user.getPassword().equals(password)
).findFirst();
if(!authenticatedUser.isPresent()){
throw new BadCredentialsException("Some Text");
}
List<GrantedAuthority> authorities = new ArrayList<>();
authorities.add(new
SimpleGrantedAuthority(authenticatedUser.get().getRole()));
Authentication auth = new UsernamePasswordAuthenticationToken(name,
password, authorities);
return auth;
}
@Override
public boolean supports(Class<?> aClass) {
return aClass.equals(UsernamePasswordAuthenticationToken.class);
}
}
```

//////////////////////////////////////////   MainController.java :

```
package com.example;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
@RestController
public class MainController {
@RequestMapping("/")
public String hello(){
return "Hello World";
}
@RequestMapping("/protected")
public String protectedHello(){
return "Hello World, i was protected";
}
@RequestMapping("/admin")
public String admin(){
return "Hello World from admin";
}
}
SpringSecurityApplication.java :
package com.example;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;
@SpringBootApplication
@ComponentScan
public class SpringSecurityApplication {
public static void main(String[] args) {
SpringApplication.run(SpringSecurityApplication.class, args);
}
}
```

//////////////////////////////////////////SpringSecurityConfig.java :

```
package com.example;
Import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builde
rs.AuthenticationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSecu
rity;
import
org.springframework.security.config.annotation.web.configuration.Ena
bleWebSecurity;
import static
org.springframework.security.config.Customizer.withDefaults;
@EnableWebSecurity
@Configuration
public class SpringSecurityConfig {
@SuppressWarnings("deprecation")
protected void configure(HttpSecurity http) throws Exception {
http.httpBasic(withDefaults()).authorizeRequests(requests ->
requests
.requestMatchers("/").permitAll()
.requestMatchers("/protected").hasRole("USER")
.requestMatchers("/admin").hasRole("ADMIN"));
}
protected void configure(AuthenticationManagerBuilder auth) throws
Exception {
auth.authenticationProvider(new CustomAuthenticationProvider());
}
}
```

////////////////////////////////////////////////////////User.java :

```
package com.example;
public class User {
private String name;
private String password;
private String role;
public User(String name, String password, String role) {
this.name = name;
this.password = password;
this.role = role;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
public String getRole() {
return role;
}
public void setRole(String role) {
this.role = role;
}
}
SpringSecurityApplicationTests.java :
package com.example;
import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;
@SpringBootTest
class SpringSecurityApplicationTests {
@Test
void contextLoads() {
}
}
```