**Implement Spring Security with Authentication**

**Screenshot**

**DemoController.java**

package com.example.demo;

import org.springframework.stereotype.Controller;

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

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

@Controller

public class DemoController {

@ResponseBody

@RequestMapping("/")

public String demo() {

return "Hi simplilearn users";

}

@ResponseBody

@RequestMapping("/admin")

public String admin() {

return "Hi admin users";

}

@ResponseBody

@RequestMapping("/user")

public String user() {

return "Hi users";

}

}

**application.properties**

server.port=8083

**Securityconfig.java**

package com.example.demo;

import org.springframework.context.annotation.Bean;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.password.NoOpPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

@EnableWebSecurity

public class Securityconfig extends WebSecurityConfigurerAdapter{

protected void configure(AuthenticationManagerBuilder auth) throws Exception {

auth.inMemoryAuthentication()

.withUser("admin")

.password("admin")

.roles("ADMIN")

.and()

.withUser("user")

.password("user")

.roles("USER");

}

@Bean

public PasswordEncoder getpassword() {

return NoOpPasswordEncoder.getInstance();

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http.authorizeRequests()

.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasAnyRole("ADMIN","USER")

.antMatchers("/").permitAll().and().formLogin();

}

}