1

package com.sprin.security1;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HomeResource {  
  
 @GetMapping("/")  
 public String home() {  
 return ("<h1>Welcome</h1>");  
 }  
  
 @GetMapping("/user")  
 public String user() {  
 return ("<h1>Welcome User</h1>");  
 }  
  
 @GetMapping("/admin")  
 public String admin() {  
 return ("<h1>Welcome Admin</h1>");  
 }  
}

package com.sprin.security1;  
  
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 SecurityConfiguration extends WebSecurityConfigurerAdapter {  
 @Override  
 protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
 // Set your configuration on the auth object  
 auth.inMemoryAuthentication()  
 .withUser("nishu")  
 .password("super")  
 .roles("USER")  
 .and()  
 .withUser("nani")  
 .password("thop")  
 .roles("ADMIN");  
 }  
  
 @Bean  
 public PasswordEncoder getPasswordEncoder() {  
  
 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();  
 }  
}

package com.sprin.security1;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringBootSecurityApplication {  
  
 public static void main(String[] args) {  
  
 SpringApplication.*run*(SpringBootSecurityApplication.class, args);  
 }  
  
}

2

package com.sprin.security2;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HomeResource {  
  
 @GetMapping("/")  
 public String home() {  
 return ("<h1>Welcome</h1>");  
 }  
  
 @GetMapping("/user")  
 public String user() {  
 return ("<h1>Welcome User</h1>");  
 }  
  
 @GetMapping("/admin")  
 public String admin() {  
 return ("<h1>Welcome Admin</h1>");  
 }  
}

package com.sprin.security2;  
  
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.bcrypt.BCryptPasswordEncoder;  
import org.springframework.security.crypto.password.PasswordEncoder;  
  
@EnableWebSecurity  
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {  
 @Override  
 protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
 // Set your configuration on the auth object  
 auth.inMemoryAuthentication()  
 .withUser("blah")  
 .password("blah")  
 .roles("USER")  
 .and()  
 .withUser("foo")  
 .password("foo")  
 .roles("ADMIN");  
 }  
  
 @Bean  
 public PasswordEncoder getPasswordEncoder() {  
  
 return new BCryptPasswordEncoder();  
 }  
  
 @Override  
 protected void configure(HttpSecurity http) throws Exception {  
 http.authorizeRequests()  
 .antMatchers("/admin").hasRole("ADMIN")  
 .antMatchers("/user").hasAnyRole("ADMIN", "USER")  
 .antMatchers("/").permitAll()  
 .and().formLogin().permitAll();  
  
 }  
}

package com.sprin.security2;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringBootSecurityApplication {  
  
 public static void main(String[] args) {  
  
 SpringApplication.*run*(SpringBootSecurityApplication.class, args);  
 }  
  
}

3

package com.sprin.security3;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
@RestController  
public class HomeResource {  
  
 @GetMapping("/")  
 public String home() {  
 return ("<h1>Welcome</h1>");  
 }  
  
 @GetMapping("/user")  
 public String user() {  
 return ("<h1>Welcome User</h1>");  
 }  
  
 @GetMapping("/admin")  
 public String admin() {  
 return ("<h1>Welcome Admin</h1>");  
 }  
}

package com.sprin.security3;  
  
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.bcrypt.BCryptPasswordEncoder;  
import org.springframework.security.crypto.password.PasswordEncoder;  
  
@EnableWebSecurity  
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {  
 @Override  
 protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
 // Set your configuration on the auth object  
 auth.inMemoryAuthentication()  
 .withUser("blah")  
 .password("$2a$09$E.fT2/vceQYX2Xq2vIvy9O6hX5Cs8gXT43Y.d8q8vwh0nC/QJfFj2")  
 .roles("USER")  
 .and()  
 .withUser("foo")  
 .password("$2a$09$9lq11vs9kR8HRp7cdRjCQuVNatbxcKCDd5bgS.HIh/jgh/7o5QM/q")  
 .roles("ADMIN");  
 }  
  
 @Bean  
 public PasswordEncoder getPasswordEncoder() {  
  
 return new BCryptPasswordEncoder();  
 }  
  
 @Override  
 protected void configure(HttpSecurity http) throws Exception {  
 http.authorizeRequests()  
 .antMatchers("/admin").hasRole("ADMIN")  
 .antMatchers("/user").hasAnyRole("ADMIN", "USER")  
 .antMatchers("/").permitAll()  
 .and().formLogin()  
 .permitAll()  
 .and()  
 .logout()  
 .permitAll()  
 .and()  
 .rememberMe()  
 .key("abcdEFGHijklmnOpqrstUVWXyz012345")  
 .tokenValiditySeconds(24\*60\*60);  
  
 }  
}

package com.sprin.security3;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class SpringBootSecurityApplication {  
  
 public static void main(String[] args) {  
  
 SpringApplication.*run*(SpringBootSecurityApplication.class, args);  
 }  
  
}