//controlelr

@RestController

**public** **class** SecurityController {

@GetMapping("/user/hello")

**public** String sayHello()

{

**return** "hello user";

}

@GetMapping("/admin/hello")

**public** String sayHello1()

{

**return** "hello admin";

}

}

//In memory authentication

@Configuration

@EnableWebSecurity

**public** **class** MySecurityConfiguration **extends** WebSecurityConfigurerAdapter{

@Bean

**public** PasswordEncoder passwordEncoder()

{

**return** **new** BCryptPasswordEncoder();

}

//authentication

**public** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception

{

String userPassword = passwordEncoder().encode("user1");

String adminPassword = passwordEncoder().encode("admin123");

System.***out***.println(userPassword);

System.***out***.println(adminPassword);

auth

.inMemoryAuthentication()

.withUser("user1")

.password(userPassword) // encrypted

.roles("USER")

.and()

.withUser("admin")

.password(adminPassword)

.roles("ADMIN");

} //authorization

**public** **void** configure(HttpSecurity httpSecurity) **throws** Exception

{

httpSecurity

.authorizeRequests()

.antMatchers("/user/\*\*")

//.hasRole("USER")

.hasAnyRole("USER","ADMIN")

.and()

.authorizeRequests()

.antMatchers("/admin/\*\*")

.hasRole("ADMIN")

.anyRequest()

.authenticated()

.and()

.formLogin();

}

}

// with custom form

**public** **void** configure(HttpSecurity httpSecurity) **throws** Exception

{

httpSecurity

.authorizeRequests()

.antMatchers("/", "/home").permitAll()

.and()

.authorizeRequests()

.antMatchers("/user/\*\*")

//.hasRole("USER")

.hasAnyRole("USER","ADMIN")

.and()

.authorizeRequests()

.antMatchers("/admin/\*\*")

.hasRole("ADMIN")

.anyRequest()

.authenticated()

.and()

.formLogin()

.loginPage("/login")

// .failureUrl(null)

.permitAll();

}

Using Thymeleaf

@Configuration

**public** **class** MvcConfig **implements** WebMvcConfigurer {

**public** **void** addViewControllers(ViewControllerRegistry registry) {

registry.addViewController("/home").setViewName("home");

registry.addViewController("/").setViewName("home");

registry.addViewController("/hello").setViewName("hello");

registry.addViewController("/login").setViewName("login");

}

}

Db authentication

@Entity

@Table(name="user1")

@AllArgsConstructor

@NoArgsConstructor

@Setter

@Getter

@Data

**public** **class** MyUser {

@Id

@Column(name="username",length = 100)

**private** String username;

**private** String password;

**private** **boolean** active;

**private** String roles;

}

@Repository

**public** **interface** UserRepo **extends** JpaRepository<MyUser, String> {

}

//application.properties

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

spring.datasource.username=root

spring.datasource.password=root

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect

spring.jpa.hibernate.ddl-auto=create

spring.sql.init.mode=always

spring.jpa.defer-datasource-initialization=true

//data.sql

insert into user1(username,password,active,roles) values('user1','user123',true,'ROLE\_USER')

insert into user1(username,password,active,roles) values('admin','admin123',true,'ROLE\_ADMIN')

insert into user1(username,password,active,roles) values('user2','user2',true,'ROLE\_USER,ROLE\_ADMIN')

//UserDetailsService

@Service

**public** **class** MyUserDetailsService **implements** UserDetailsService {

@Autowired

**private** UserRepo userRepo;

@Override

**public** UserDetails loadUserByUsername(String username) **throws** UsernameNotFoundException {

//System.out.println("Username Entered.."+username);

MyUser user=**null**;

Optional<MyUser> opUser = userRepo.findById(username); // from db

**if**(opUser.isPresent())

user = opUser.get();

//else

// throw new UsernameNotFoundException("error");

/\*

\* ArrayList<SimpleGrantedAuthority> list = new ArrayList<>(); String users[] =

\* user.getRoles().split(","); // "ROLE\_USER,ROLE\_ADMIN"

\* System.out.println(user.getRoles());

\*

\* for(String s:users) { list.add(new SimpleGrantedAuthority(s)); }

\*/

//return new MyUserDetails(user); // option 1

/\* option 2 \*/

System.***out***.println("Roles:"+user.getPassword()+" "+user.getRoles());

List<GrantedAuthority> list ;

list= Arrays.*stream*(user.getRoles().split(",")).map(SimpleGrantedAuthority::**new**).collect(Collectors.*toList*());

**return** **new** User(user.getUsername(), user.getPassword(), list);

}

}

// security configuration

@Configuration

@EnableWebSecurity

**public** **class** MySecurityConfigurationDb **extends** WebSecurityConfigurerAdapter {

@Autowired

**private** MyUserDetailsService userDetailsService;

/\*

\* does not work in mysql

\*

\* @Bean public PasswordEncoder passwordEncoder() { return new

\* BCryptPasswordEncoder(); }

\*/

@Bean

**public** PasswordEncoder passwordEncoder() {

PasswordEncoder encoder = **new** PasswordEncoder()

{

@Override

**public** String encode(CharSequence userEntered) {

**return** userEntered.toString(); }

@Override

**public** **boolean** matches(CharSequence userEntered, String password) {

System.***out***.println("UserEntered "+userEntered);

System.***out***.println("password"+password);

**if**(password.equals(userEntered))

**return** **true**;

**return** **false**;

} };

**return** encoder;

}

**public** **void** configure(AuthenticationManagerBuilder auth) **throws** Exception

{

auth.userDetailsService(userDetailsService)

/\* calls loadUserByUsername(String username), returns the User object with other values \*/

.passwordEncoder(passwordEncoder());

}

**public** **void** configure(HttpSecurity httpSecurity) **throws** Exception

{

httpSecurity

.authorizeRequests()

.antMatchers("/user/\*\*")

//.hasRole("USER")

.hasAnyRole("USER","ADMIN")

.and()

.authorizeRequests()

.antMatchers("/admin/\*\*")

.hasRole("ADMIN")

.anyRequest()

.authenticated()

.and()

.formLogin();

}

}