References

|  |  |
| --- | --- |
| spr security all version docs | <https://docs.spring.io/spring-security/reference/servlet/authentication/passwords/index.html> |
| spring security home | <https://docs.spring.io/spring-boot/docs/1.3.8.RELEASE/reference/html/boot-features-security.html> |
|  |  |

Spring Security updates

|  |  |
| --- | --- |
| Here after we should not write a class extending WebSecurityConfigurerAdapter | from spring 6 onwards we should use a “SecurityFilterChain” bean to configure HttpSecurity or  configure “WebSecurityCustomizer” bean to configure WebSecurity |
| add this jar  Note:- when u add this jar by default spring sec is enabled  default user name is user  default pass will be printed on console  while hitting a web service you have to pass this credentials under auth type basic auth | *<!-- if u un comment this security by default u have to pass credentials in postman basic auth tab default user name is user, default pass will be printed on console -->*  *<dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-security</artifactId>  </dependency>*  if u don’t want the default user name and password u can configure below properties- so that u have to pass these in postman basic auth tab while hitting a web service  *spring.security.user.name=customuser spring.security.user.password=custompassword*  *incase if csrf is enabled by default, u have to disabled using below config*  *@Configuration @EnableWebSecurity public class MySprSecConfig {  @Bean  public SecurityFilterChain filterChain(HttpSecurity httpSecurity){  return httpSecurity.authorizeHttpRequests().anyRequest()  .authenticated().and().httpBasic()  .and().csrf().disable().build();  } }* |
|  |  |

|  |  |
| --- | --- |
| The below code works on spring boot 2.7 only | public SecurityFilterChain configure(HttpSecurity http) throws Exception {  return http  .csrf().disable()  .authorizeRequests().  antMatchers("/").permitAll()  .antMatchers("/user").hasRole("USER")  .antMatchers("/admin").hasRole("ADMIN")  .anyRequest().authenticated()  .and()  .httpBasic()  .and().build();  } |

Inmemory auth filter

|  |  |
| --- | --- |
| @Configuration  @EnableWebSecurity  public class SecurityConfig {  @Bean  public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {  http  .authorizeHttpRequests((authorize) -> authorize  .anyRequest().authenticated()  )  .httpBasic(Customizer.withDefaults())  .formLogin(Customizer.withDefaults());  return http.build();  }  @Bean  public UserDetailsService userDetailsService() {  UserDetails userDetails = User.withDefaultPasswordEncoder()  .username("user")  .password("password")  .roles("USER")  .build();  return new InMemoryUserDetailsManager(userDetails);  }  } |  |