1.spring security配置

示例：

让spring发现它

//AbstractSecurityWebApplicationInitializer实现了WebApplicationInitializer  
 //因此spring会发现它，并用它在Web容器中注册DelegatingFilterProxy  
 //可以重载它的appendFilters或insertFilters方法来注册自己选择的filter  
public class SecurityWebInitializer extends AbstractSecurityWebApplicationInitializer {  
}

Security配置类

@Configuration  
@ComponentScan  
@EnableWebSecurity //启动Web安全性  
public class SecurityConfig extends WebSecurityConfigurerAdapter{  
 @Autowired  
 private DataSource dataSource;  
 //配置如何通过拦截器保护请求  
 @Override  
 protected void configure(HttpSecurity http) throws Exception {  
http.authorizeRequests().anyRequest().authenticated().and().formLogin().and().httpBasic();  
 }  
 //配置认证用户列表  
 @Override  
 protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
 auth.jdbcAuthentication().dataSource(dataSource).  
 usersByUsernameQuery("select name,password,true from tbl\_user where name=?").  
 authoritiesByUsernameQuery("select name,ROLE\_USER from tbl\_user where name=?");  
 }  
}

其中configure(WebSecurity) 方法可以配置filter链

扩展：

1. 拦截所有请求

http.authorizeRequests().anyRequest().authenticated().and().formLogin().and().httpBasic();

1. 拦截指定uri

http.authorizeRequests().antMatchers("/user/\*").authenticated()  
 .antMatchers(HttpMethod.*POST*, "/xhsf/\*").authenticated()  
 .anyRequest().permitAll();

1. 指定权限和uri

@Override  
protected void configure(HttpSecurity http) throws Exception {  
 http  
 .formLogin().and()  
 .authorizeRequests()  
 .antMatchers("/user/\*").hasRole("USER")  
 .antMatchers(HttpMethod.*POST*, "/xhsf/\*").hasRole("USER")  
 .anyRequest().permitAll();  
}

1. 使用http basic认证

.and().httpBasic()

1. 将最具体的请求路径放在前面，不具体的放在后面，不然会覆盖（如anyRequest放在最后面）
2. 设置rememberme功能

.and().rememberMe().tokenValiditySeconds(100).key("xhsf")