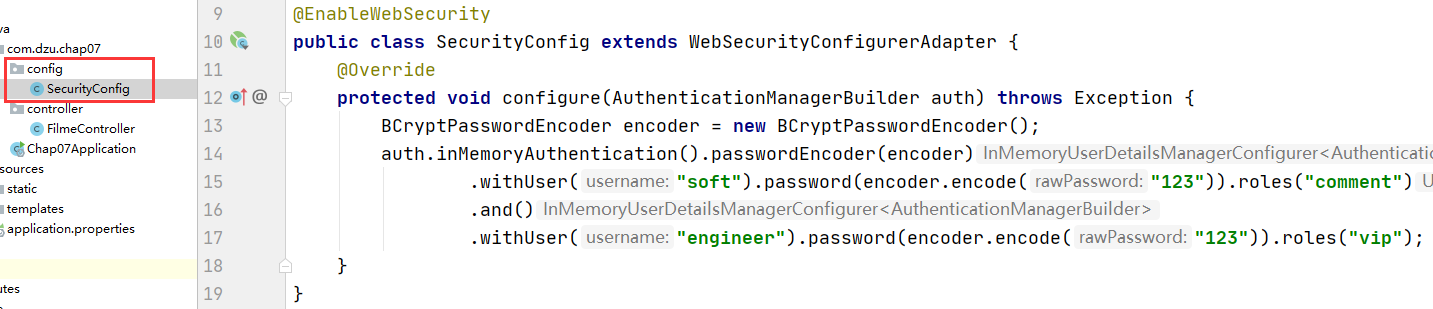
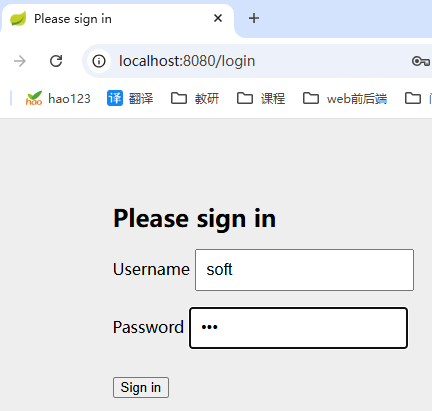
# 实验十六 自定义用户认证

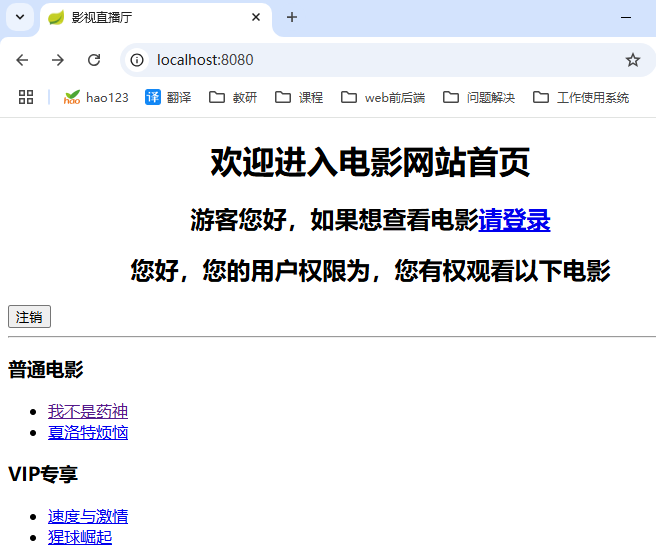
一、内存身份认证

1.自定义SecurityConfig类，继承WebSecurityConfigurerAdapter类



2.测试





二、JDBC身份认证

1.数据准备

DROP TABLE IF EXISTS `t\_customer`;

CREATE TABLE `t\_customer` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(200) DEFAULT NULL,

`password` varchar(200) DEFAULT NULL,

`valid` tinyint(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of t\_customer

-- ----------------------------

INSERT INTO `t\_customer` VALUES ('1', 'shitou', '$2a$10$Pv69045c6Zf/8pDGvKqwZ.6HOqizgL9/LvxzoVvWjTlx4lIeYI9q2 ', '1');

INSERT INTO `t\_customer` VALUES ('2', '李四', '$2a$10$Pv69045c6Zf/8pDGvKqwZ.6HOqizgL9/LvxzoVvWjTlx4lIeYI9q2 ', '1');

-- ----------------------------

-- Table structure for t\_authority

-- ----------------------------

DROP TABLE IF EXISTS `t\_authority`;

CREATE TABLE `t\_authority` (

`id` int(20) NOT NULL AUTO\_INCREMENT,

`authority` varchar(20) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of t\_authority

-- ----------------------------

INSERT INTO `t\_authority` VALUES ('1', 'role\_common');

INSERT INTO `t\_authority` VALUES ('2', 'role\_vip');

-- ----------------------------

-- Table structure for `t\_customer\_authority`

-- ----------------------------

DROP TABLE IF EXISTS `t\_customer\_authority`;

CREATE TABLE `t\_customer\_authority` (

`id` int(20) NOT NULL AUTO\_INCREMENT,

`customer\_id` int(20) DEFAULT NULL,

`authority\_id` int(20) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=3 DEFAULT CHARSET=utf8;

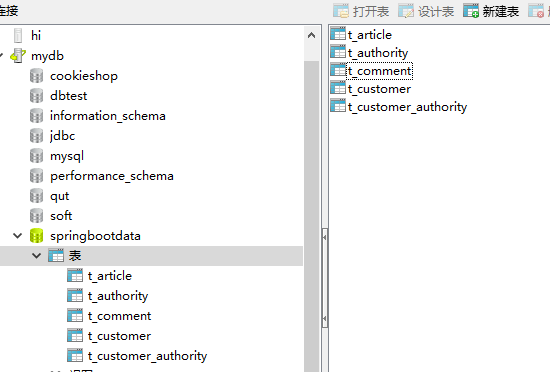
-- ----------------------------

-- Records of t\_customer\_authority

-- ----------------------------

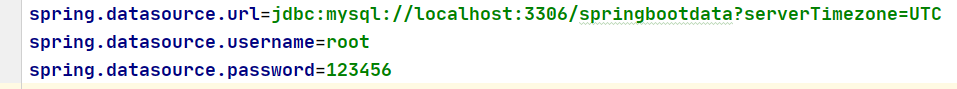
INSERT INTO `t\_customer\_authority` VALUES ('1', '1', '1');

INSERT INTO `t\_customer\_authority` VALUES ('2', '2', '2');



2.添加JDBC连接数据库的依赖启动器

3.进行数据库连接配置



spring.datasource.driver-class-name=com.mysql.jdbc.Driver  
spring.datasource.url=jdbc:mysql://localhost:3306/springbootdata?serverTimezone=UTC  
spring.datasource.username=root  
spring.datasource.password=123456

4.使用JDBC进行身份认证

@EnableWebSecurity  
public class SecurityConfig extends WebSecurityConfigurerAdapter {  
 @Autowired  
 private DataSource dataSource;  
 @Override  
 protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
 BCryptPasswordEncoder encoder = new BCryptPasswordEncoder();  
 String userSQL ="select username,password,valid from t\_customer " +  
 "where username = ?";  
 String authoritySQL="select c.username,a.authority from t\_customer c,t\_authority a,"+  
 "t\_customer\_authority ca where ca.customer\_id=c.id " +  
 "and ca.authority\_id=a.id and c.username =?";  
 auth.jdbcAuthentication().passwordEncoder(encoder)  
 .dataSource(dataSource)  
 .usersByUsernameQuery(userSQL)  
 .authoritiesByUsernameQuery(authoritySQL);  
 System.*out*.println("密码:"+encoder.encode("123"));  
 }  
}

