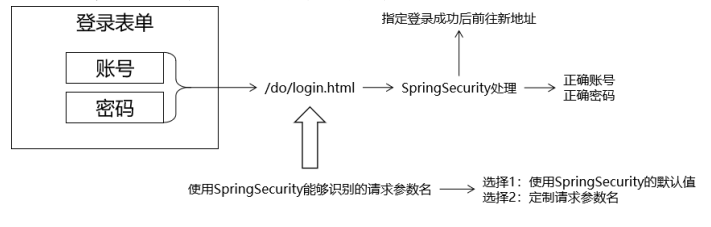
## 登录



登录表单提交后，SpringSecurity要获取到 账号和密码 去验证，有两种获取方式

1. 默认账号的请求参数名：username 默认密码的请求参数名：password

2.定制自定义的 账号和密码 参数，此时需要在配置类中指定，通过usernameParameter()和 passwordParameter()方法。

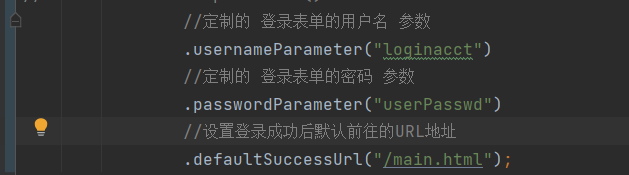
主要是保持表单项的name属性 要和 SpringSecurity的参数一致

### 配置类认证方法

WebAppSecurityConfig配置类修改：

configure(HttpSecurity security)方法中增加 自定义用户名和密码的请求参数

设置登录成功后访问的页面

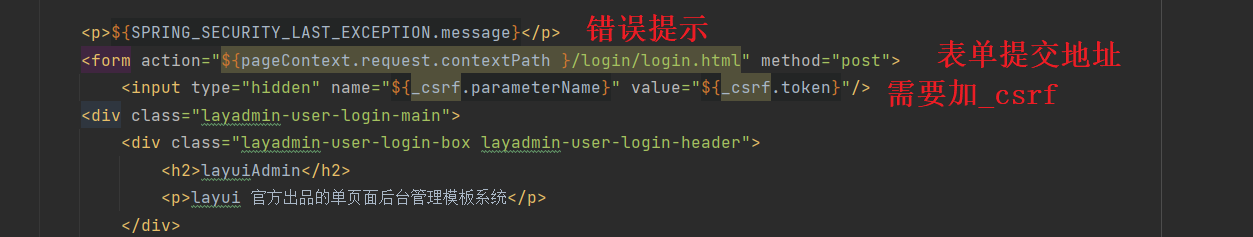


### 首页表单

index.jsp页面修改：

增加form表单

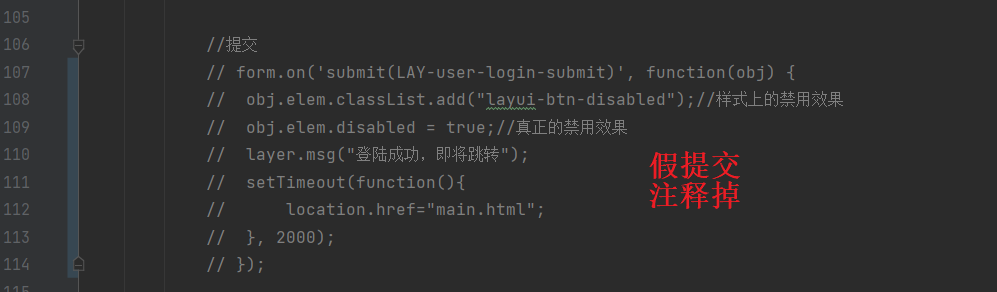
\_csrf是为了防止跨站请求伪造



请求表单的参数名



假提交，注释掉



### 配置类授权方法

3.重写配置类的另一个方法

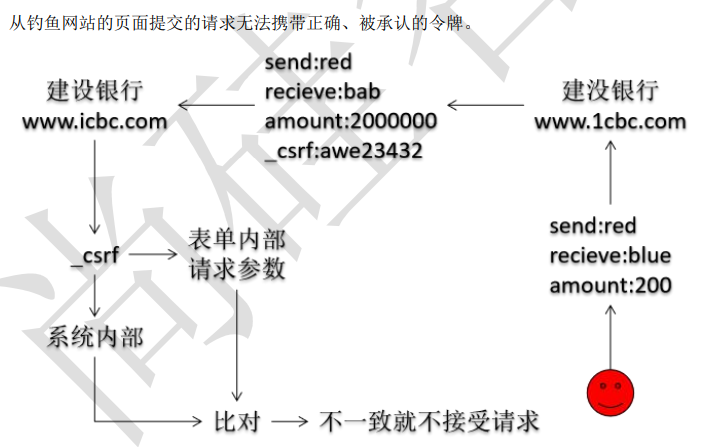
configure(AuthenticationManagerBuilder auth)

@Override  
protected void configure(AuthenticationManagerBuilder auth) throws Exception {  
  
 //角色和权限必须要有一个才能通过SpringSecurity的认证  
 auth  
 //暂时在内存中验证账号和密码  
 .inMemoryAuthentication()  
 //设置账号密码  
 .withUser("tom").password("123456")  
 //设置角色  
 .roles("ADMIN")  
 .and()  
 .withUser("andy").password("123456")  
 //设置权限  
 .authorities("UPDATE");  
}

Cannot pass a null GrantedAuthority collection 问 题 是 由于没有设置roles() 或authorities()方法导致的。

实现的最后效果：登录成功后具体资源都可以访问了。

### 跨站请求伪造csrf



## 注销

通过调用 HttpSecurity 对象的一系列方法设置注销功能。

logout()方法：开启注销功能

logoutUrl()方法：自定义注销功能的 URL 地址

如果 CSRF 功能没有禁用，那么退出请求必须是 POST 方式。如果禁用了CSRF 功能则任何请求方式都可以。

logoutSuccessUrl()方法：退出成功后前往的 URL 地址

addLogoutHandler()方法：添加退出处理器

logoutSuccessHandler()方法：退出成功处理器

# 最后