

[**#**](https://pig4cloud.com/#%E4%B8%BA%E4%BB%80%E4%B9%88%E8%A6%81%E5%81%9A%E4%BA%8C%E6%AC%A1%E8%AE%A4%E8%AF%81) **为什么要做二次认证**

spring boot admin 默认没有开启认证，也是就是别人知道了监控模块的IP:PORT 即可访问。监控功能在生产上又是必要的功能，所以需要有二次认证

[**#**](https://pig4cloud.com/#%E5%AE%9E%E7%8E%B0%E5%8E%9F%E7%90%86) **实现原理**

* 引入spring security

<!--security-->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

* 配置spring security即可

@Configuration

public class WebSecurityConfigurer extends WebSecurityConfigurerAdapter {

private final String adminContextPath;

public WebSecurityConfigurer(AdminServerProperties adminServerProperties) {

this.adminContextPath = adminServerProperties.getContextPath();

}

@Override

protected void configure(HttpSecurity http) throws Exception {

// @formatter:off

SavedRequestAwareAuthenticationSuccessHandler successHandler = new SavedRequestAwareAuthenticationSuccessHandler();

successHandler.setTargetUrlParameter("redirectTo");

successHandler.setDefaultTargetUrl(adminContextPath + "/");

http

.headers().frameOptions().disable()

.and().authorizeRequests()

.antMatchers(adminContextPath + "/assets/\*\*"

, adminContextPath + "/login"

, adminContextPath + "/actuator/\*\*"

).permitAll()

.anyRequest().authenticated()

.and()

.formLogin().loginPage(adminContextPath + "/login")

.successHandler(successHandler).and()

.logout().logoutUrl(adminContextPath + "/logout")

.and()

.httpBasic().and()

.csrf()

.disable();

// @formatter:on

}

}

* 在对应的 pig-monitor-dev.yml 配置用户

pig 默认的登录用户 pig/pig，可以参考配置文件加解密章节

spring:

security:

user:

name: ENC(8Hk2ILNJM8UTOuW/Xi75qg==) # pig

password: ENC(o6cuPFfUevmTbkmBnE67Ow====) # pig