



## @PreAuthorize/@PostAuthorize

@PreAuthorize 和 @PostAuthorize 都是方法或类级别注解。

```
@Target({ ElementType.METHOD, ElementType.TYPE })
@Retention(RetentionPolicy.RUNTIME)
@Inherited
@Documented
public @interface PreAuthorize {

    /**
     * @return the Spring-EL expression to be evaluated before invoking the protected
     * method
     */
    String value();
}
```

- @PreAuthorize 表示访问方法或类在执行之前先判断权限，大多情况下都是使用这个注解，注解的参数和 access() 方法参数取值相同，都是权限表达式。
- @PostAuthorize 表示方法或类执行结束后判断权限，此注解很少被使用到。

### 开启注解

```
@SpringBootApplication
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class SpringsecurityDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringsecurityDemoApplication.class, args);
    }

}
```

### 添加@PreAuthorize

在控制器方法上添加 @PreAuthorize，参数可以是任何 access() 支持的表达式

```
/**
 * 成功后跳转页面
 * @return
 */
@PreAuthorize("hasRole('ROLE_abc')")
@RequestMapping("/toMain")
public String toMain(){
    return "redirect:/main.html";
}
```