基于注解的访问控制

在 Spring Security 中提供了一些访问控制的注解。这些注解都是默认是都不可用的，需要通过 `@EnableGlobalMethodSecurity` 进行开启后使用。

如果设置的条件允许，程序正常执行。如果不允许会报 500

```
org.springframework.security.access.AccessDeniedException: 不允许访问
```

这些注解可以写到 Service 接口或方法上，也可以写到 Controller或 Controller 的方法上。通常情况下都是写在控制器方法上的，控制接口URL是否允许被访问。

@Secured

@Secured 是专门用于判断是否具有角色的。能写在方法或类上。参数要以 ROLE_开头。

```
@Target({ ElementType.METHOD, ElementType.TYPE })
@Retention(RetentionPolicy.RUNTIME)
@Inherited
@Documented
public @interface Secured {
    /**
     * Returns the list of security configuration attributes (e.g. &amp;nbsp;ROLE_USER, ROLE_ADMIN).
     *
     * @return String[] The secure method attributes
     */
    public String[] value();
}
```

开启注解

在启动类(也可以在配置类等能够扫描的类上)上添加 `@EnableGlobalMethodSecurity(securedEnabled = true)`

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

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

}
```

在控制器方法上添加@Secured 注解

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

配置类

```
@Override
protected void configure(HttpSecurity http) throws Exception {
   //表单提交
   http.formLogin()
         //自定义登录页面
         .loginPage("/login.html")
         //当发现/login时认为是登录，必须和表单提交的地址一样。去执行UserServiceImpl
         .loginProcessingUrl("/login")
         //登录成功后跳转页面，POST请求
         .successForwardUrl("/toMain")

   //url拦截
   http.authorizeRequests()
         //login.html不需要被认证
         .antMatchers("/login.html").permitAll()
         //所有请求都必须被认证，必须登录后被访问
         .anyRequest().authenticated();
   //关闭csrf防护
   http.csrf().disable();
}
```