# **SpringBoot 常用注解简单说明** <https://blog.csdn.net/fxbin123/article/details/80387668>

@SpringBootApplication \*Application的入口类 核心注解

@SpringBootConfiguration 表示Application作为配置文件存在

@EnableAutoConfiguration 表示启用SpringBoot内置的自动配置功能

@ComponentScan 路径为Application类所在package以及package下的子路径  
  
@Controller 处理 HTTP 请求

@RestController = @Controller +@ResponseBody  
  
@RequestMapping 配置url映射（url路径）  
@GetMapping+@PostMapping  
  
@ExceptionHandler(value = Exception.class) 捕获异常 （用于统一异常处理）

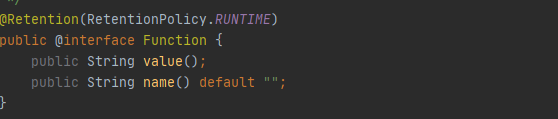
@PathVariable 获取url中的数据

@RequestParam 获取请求参数的值 ？id=100

@Pointcut 注解声明切入点

@Autowired 自动装配类  
  
@PropertySource(“classpath:xx.properties”) 加载配置文件，加载后可用Environment类getProperty调用具体配置项  
  
@Value(“配置项”) 加载配置文件之后，可直接将配置项放入类属性中（非静态）  
  
  
**SpringBoot 自定义注解说明**

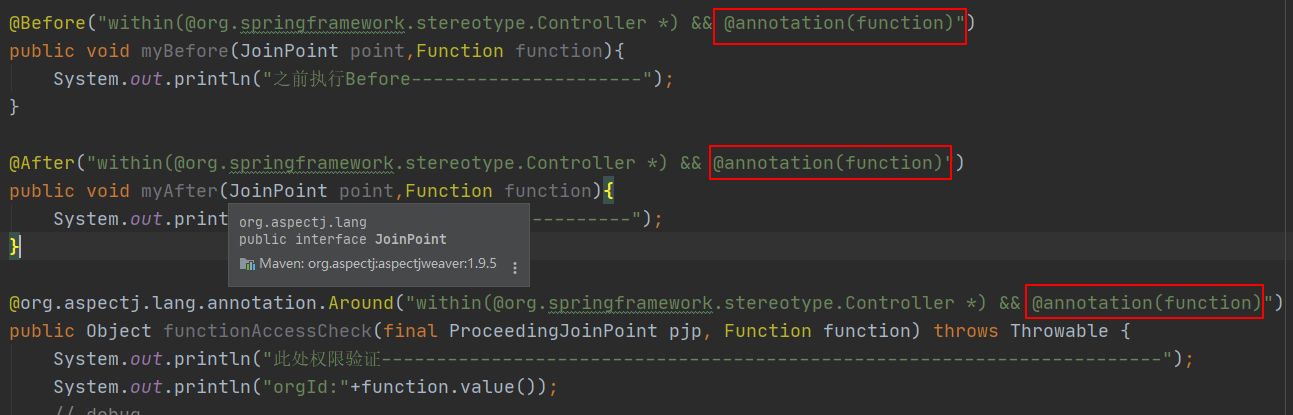
定义注解**接口类**

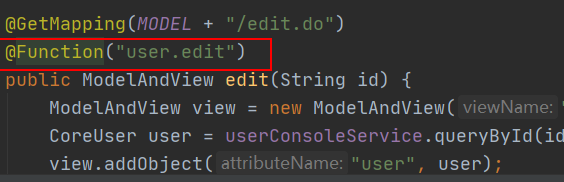


定义注解**实现类**

类名与位置无要求

重点：使用@Around、@Before、@After等注解，使用@annotation(注解接口类名)定义注解的具体实现方法

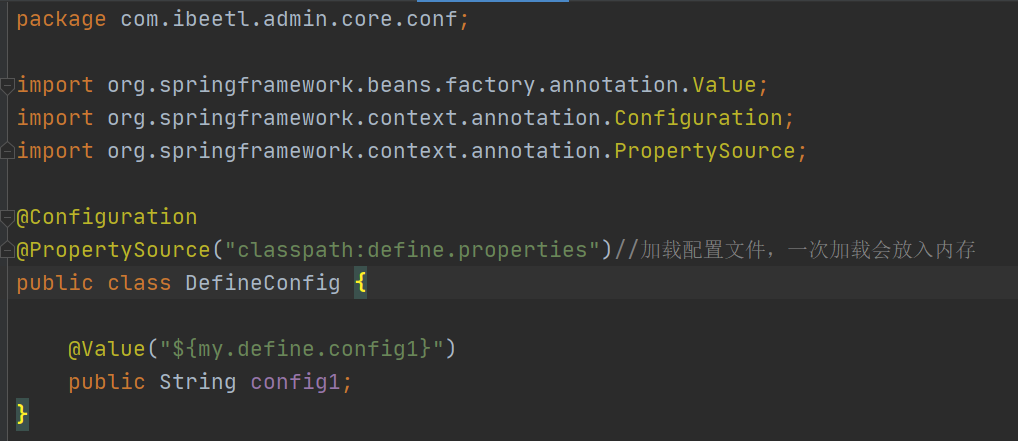
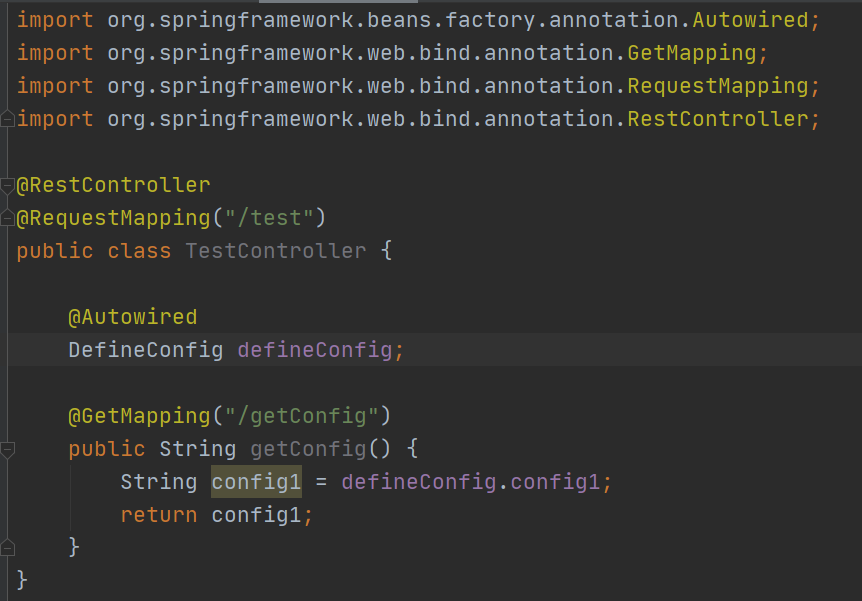


使用注解  
引入注解接口类后即可直接使用

**SpringBoot 自定义配置文件说明  
Application.properties为spring boot自动配置类，可直接使用（Environment类getProperty调用具体配置项）**

自定义配置文件define.properties



自定义配置类，并使用@PropertySource引入配置文件，使用@Value将配置项放入类属性  
  
使用配置，@Autowired装配配置类之后，可直接调用属性获取配置  
  
ps：也可不创建配置类，使用@PropertySource引入配置文件后，使用**Environment类getProperty直接调用具体配置项；**

**使用配置类的好处，修改配置文件（配置项名称等）之后，无需修改使用配置项的类，只需修改相应的配置类**