完整的注解需要三个方面

1、需要定义一个注解 @Interface

2、需要将注解放在程序中

3、注解解析赋予注解功能

jdk中的元注解

用在注解上的注解。元注解主要用于限定当前的注解能贴在哪里？能保留在哪个阶段（源代码，字节码，运行时（jvm中））

@Retention 表示注解保存在哪一个阶段。三个取值source，class，runtime保存在RetentionPolicy枚举中。

source：注解保留到字节码阶段，运行时就失效了

runtime：注解保留到运行阶段，运行时候使用反射获取做相应的程序处理-一般开发者自定注解都保留运行阶段

source：注解在源代码阶段有效，编译字节码就失效了

@Target 表示注解可以贴在哪里。位置常量ElementType中

ElementType.ANNOTATION\_TYPE只能修饰Annotation

ElementType.CONSTRUCTOR只能修饰构造方法

ElementType.FIELD只能修饰字段(属性),包括枚举常量

ElementType.LOCAL\_VARIABLE只能修饰局部变量

ElementType.METHOD只能修饰方法

ElementType.PACKAGE只能修饰包(极少使用)

ElementType.PARAMETER只能修饰参数

ElementType.TYPE只能修饰类，接口，枚举

运行时反射，比较耗时。

Springweb中拦截器HandlerInterceptor可以用于检查ip黑白名单

Feign中也有一个拦截器RequestInterceptor

@Document表示 将注解包含到JavaDoc中

@Inherited允许子类继承父类的注解。