J2SE5.0版本在 java.lang.annotation提供了四种元注解，专门注解其他的注解：

@Documented –注解是否将包含在JavaDoc中  
@Retention –什么时候使用该注解  
@Target? –注解用于什么地方  
@Inherited – 是否允许子类继承该注解

例如本项目使用@sysLog,然后通过切面记录日志

@Aspect

@Component

**public** **class** SysLogAspect {

@Autowired

**private** SysLogService sysLogService;

@Pointcut("@annotation(com.timesaving.annotation.SysLog)")

**public** **void** logPointCut() {

}

@Before("logPointCut()")

**public** **void** saveSysLog(JoinPoint joinPoint) {

MethodSignature signature = (MethodSignature) joinPoint.getSignature();

Method method = signature.getMethod();

SysLogEntity sysLog = **new** SysLogEntity();

SysLog syslog = method.getAnnotation(SysLog.**class**);

**if** (syslog != **null**) {

// 注解上的描述

sysLog.setOperation(syslog.value());

}

// 请求的方法名

String className = joinPoint.getTarget().getClass().getName();

String methodName = signature.getName();

sysLog.setMethod(className + "." + methodName + "()");

// 请求的参数

Object[] args = joinPoint.getArgs();

String params = JSON.*toJSONString*(args[0]);

sysLog.setParams(params);

// 获取request

HttpServletRequest request = HttpContextUtils.*getHttpServletRequest*();

// 设置IP地址

sysLog.setIp(IPUtils.*getIpAddr*(request));

// 用户名

String username = ShiroUtils.*getUserEntity*().getUsername();

sysLog.setUsername(username);

sysLog.setCreateDate(**new** Date());

// 保存系统日志

sysLogService.save(sysLog);

}

}