**软件日志测试方法总结**

　　一.我们为什么要做[日志测试](javascript:;" \t "_self)？

　　虽然额外的日志[记录](javascript:;" \t "_self)可能会导致软件程序的执行速度下降。但如果有详细的日志记录就可以帮助我们更迅速地诊断问题，加快我们对故障的响应，并且往往可以显著地减少发现一些隐藏得非常深的错误的问题！

　　二.日志级别

　　log4j定义了8个级别的log（除去OFF和ALL，可以说分为6个级别），优先级从高到低依次为：OFF、FATAL、ERROR、WARN、INFO、DEBUG、TRACE、 ALL。

　　ALL 最低等级的，用于打开所有日志记录。

　　TRACE designates finer-grained informational events than the DEBUG.Since:1.2.12，很低的日志级别，一般不会使用。

　　DEBUG 指出细粒度信息事件对调试应用程序是非常有帮助的，主要用于开发过程中打印一些运行信息。

　　INFO 消息在粗粒度级别上突出强调应用程序的运行过程。打印一些你感兴趣的或者重要的信息，这个可以用于生产环境中输出程序运行的一些重要信息，但是不能滥用，避免打印过多的日志。

　　WARN 表明会出现潜在错误的情形，有些信息不是错误信息，但是也要给程序员的一些提示。

　　ERROR 指出虽然发生错误事件，但仍然不影响系统的继续运行。打印错误和异常信息，如果不想输出太多的日志，可以使用这个级别。

　　FATAL 指出每个严重的错误事件将会导致应用程序的退出。这个级别比较高了。重大错误，这种级别你可以直接停止程序了。

　　OFF 最高等级的，用于关闭所有日志记录。

　　如果将log level设置在某一个级别上，那么比此级别优先级高的log都能打印出来。一般来说测试环境会设置为DEBUG级别，客户正式环境会设置为ERROR级别。

　　三.如何查看日志？

　　windows的直接使用Notepad++、UE编辑器查看日志文件就可以了，以下详细介绍linux系统的日志查看方法：

　　1.使用ssh工具连接服务器

　　2.连接成功后，使用cd命令切换到tomcat/logs/下的catalina.out文件

　　3.使用tail命令查看日志文件，例如tail -f catalina.out，默认会显示最后10行日志，-f参数可以循环读取日志，当前产生的所有日志都会输出。如果需要查看日志文件中已存在的N行内容，可以使用命令tail -f -n 1000 catalina.out（查看文件中1000行内容，并循环输出当前产生的日志）

　　4.一些偶发的问题，我们在提交bug时，可以将错误堆栈截图上传，帮助开发更快的定位问题。