

LOGBack : Java日志的新进展 PDF转换可能丢失图片或格式 , 建议阅读原文

[https://www.100test.com/kao\\_ti2020/290/2021\\_2022\\_LOGBack\\_EF\\_BC\\_9A\\_c104\\_290405.htm](https://www.100test.com/kao_ti2020/290/2021_2022_LOGBack_EF_BC_9A_c104_290405.htm) Ceki G ü lc ü 在Java日志领域世界知名。他创造了Log4J，这个最早的Java日志框架即便在JRE内置日志功能的竞争下仍然非常流行。随后他又着手实现SLF4J这个“简单的日志前端接口（Facade）”来替代Jakarta Commons-Logging。在过去的一年中，Ceki在从事他的新项目，LOGBack，一个“可靠、通用、快速而又灵活的Java日志框架”。自一年前发布0.1 alpha版以来，LOGBack已经取得了长足的进步。1.0版即将发布，又有早期用户的正面评价，我们也应该仔细看看LOGBack，到底适不适合我们的需要。Xavier Hanin 谈论了他使用LOGBack的经验：我已经用了LOGBack几个月，我被它打动了。文档和支持都很完善，日志功能简洁利落，性能表现也说得上风驰电掣，还有创新的Eclipse插件，我终于给劳苦功高的Log4J找到了接班人。Rob Willams 补充说：噢，还有，我们当初毅然决定采用LogBack.爱死它了。整个过渡过程一点麻烦都没有，我们绝对喜欢它的新语法。他所说的新语法让LOGBack能够处理许多复杂的日志语句，而不再需要事先检查日志级别（logging level），同时性能上的影响微不足道。比如在Log4J里面，你可能会这样写：  
`if ( logger.isDebugEnabled ( ) ) { logger.debug ( "User with account " user.getAccount ( ) " failed authentication ; " "supplied crypted password " user.crypt ( password ) " does not match." ) ; }` 等价的LOGBack语句如下：`logger.debug ( "User with account {} failed authentication ; " "supplied crypted password {} does not`

match." , user.getAccount ( ) , user.crypt ( password ) ) ;

LOGBack把拼装消息的代价推迟到它能够确定是不是要显示这条消息的时候。不过获取参数的高昂代价并没有被推迟支付，比如上例中的密码加密。LOGBack还声称自己性能更佳：某些关键操作，比如判定是否记录一条日志语句的操作，其性能得到了显著的提高。这个操作在LOGBack中需要3纳秒，而在Log4J中则需要30纳秒。LOGBack创建记录器（logger）的速度也更快：13毫秒，而在Log4J中需要23毫秒。更重要的是，它获取已存在的记录器只需94纳秒，而Log4J需要2234纳秒，时间减少到了1/23.跟JUL相比的性能提高也是显著的。LOGBack还可以被集成，目前已经有了一个Eclipse插件和一个JMX Configurator Bean. InfoQ就LOGBack访问了Ceki，第一个问题是大家都关心的：为什么要建立另一个日志框架，而不是把这些改进放到Log4J中去？我当时（今天也还部分地）觉得在Apache Logging Services项目之外做创新会容易一些。不要误会我的意思，我对Apache Software Foundation的评价很高，它是一个独特的，而且在很多方面都极其出色的组织。谁也说不定，可能有一天SLF4J和LOGBack成为日志领域新的事实标准的时候，会重新融入Apache. 关于SLF4J的接受程度：现在有几个重量级项目，比如Hibernate、Jetty、Spring-OSGi和Wicket都已经迁移到了SLF4J API，我可以毫不惭愧地说，SLF4J的吸引力不可忽视。SLF4J正在四处冒出头来，虽然Jakarta Commons Logging（JCL）这个广泛使用的类库，在Apache品牌的温暖阳光沐浴下，占据着整个疆域。考虑到开始时的劣势，SLF4J现在的表现已经超过了我们的预期。当被问到如何比较LOGBack的免费文档，和Log4J那种小部分免

费大部分商业的文档时：如你所说，Log4J只提供了有限的免费文档，完善的文档则需要付费。对于LOGBack，我们采纳了另一种途径，我们所有的文档都可以在我们的项目网站上直接看到，获取也完全免费，这给了Java开发者们又一个理由转移到LOGBack.另外，LOGBack的市场占有率比Log4J要低一个数量级，销售LOGBack的文档没有任何经济上的意义。为了保证项目的长期经济支持，我们开发了一个跟LOGBack稍稍相关的产品，过几周就会推出。我们对LOGBack的长期计划是不搞任何噱头，把它发展成一个合作性的开源项目。至于“合作性”，我的意思是开方给所有开发者做贡献，而不局限于现在的开发团队。请说说离1.0版发布还差些什么：对于即将到来的1.0版，大部分重要的东西都已经齐备了。我们还要修复很多错误，但主要的工作还是完善文档，做更多的测试，一再重复做这些事情。我说过要完善文档没？在日志这个领域还有很多东西需要我们去做的，可能要好几代人才能完成。我们已经逐渐看清了前面的路途，希望能够给未来铺平一些道路。当被问到LOGBack有哪些能够吸引开发者的地方：没有绝对的答案。有些用户可能会觉得性能是一个值得转移到LOGBack的理由，其他人可能觉得Log4J已经不错。虽然我和其他LOGBack开发者们正努力给用户更多的理由，不过我们中间很多人对于能够从事一个强调质量的软件项目已经很满意。我们必须“吃我们自己的狗食”，并在这个过程中得到有价值的软件开发技能按今天的标准来说并不坏。考虑到LOGBack项目的出发点正是Log4J项目的不及之处，只要我们持续不断地改进LOGBack，我们相信一定会有越来越多的Java开发者接受SLF4J/LOGBack的组合。因为SLF4J

和LOGBack可以桥接其他竞争性的API，开发者们可以在他们的项目中用LOGBack替换Log4J（通过log4j-bridge.jar）和Jakarta Commons-Logging（通过jcl104-over-slf4j.jar），因而不必仅仅为了使用LOGBack而被迫在同一个项目中配置好几个日志框架。关于LOGBack和SLF4J的更多信息请阅读Ceki的《十个转移到LOGBack的理由》，或者持续关注InfoQ的Java社区。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)