链滴

[翻译] 我的日志最佳实践

作者：laoma

原文链接：https://ld246.com/article/1634611067788

来源网站：链滴

原文地址

我的日志最佳实践

如果您跟我一样是一个后端开发人员，那么日志则是您应用程序的窗口。与前端开发不同，后端开发了一些日志之外就没什么可看的了。

以下是我在打日志时使用的一些个人准则。

在事件发生后打日志，而不是之前

在过去，每艘船都写航海日志。它就像一本日记，记录了一天中发生的所有重要事件。

打日志就像写航海日志一样，我们应该记录已经发生的事情，而不是我们将要做的事情。

让我们举个例子：

```
// 别这么打日志
log.info("向 REST API 发送请求")
restClient.makeRequest()

// 要像这样打日志
restClient.makeRequest()
log.info("向 REST API 发送了个请求")
```

第一个日志没有说明太多内容。当看到这条日志时你不知道对REST Api的调用是否成功。为此，你必查找是否存在异常。如果你看到了这条日志但错过了后面的异常，你将在一天的剩余时间里感到困惑相信我）。

第二个日志要好得多。它清楚地表明之前的操作是成功的。如果 REST Api的调用失败，你就不会看这条日志，而是会出现异常。

我将此规则应用于所有 INFO级别的日志。但是，我为 DEBUG级别的日志设置了例外。

分隔参数和消息

一条典型的日志包括两部分内容：一是记录当前运行状态的手写消息，二是操作中涉及的（技术）参列表。

你应该尝试将两个部分分开。

```
// 别这么写
restClient.makeRequest()
log.info("向Rest Api {} 发送请求。", url)

// 要像这么写
restClient.makeRequest()
log.info("向Rest Api发送请求。 [url={}]", url)
```

第一条日志消息有一些缺陷。例如，很难按照 Grok 模式进行解析。因此，这会使我们使用日志工具日志中提取 ID 或参数变得更加困难，当然读起来也很困难。想象一下，上面的日志中那个 URL很长且末尾带有参数列表，一半的日志消息根本不在您的屏幕上。而且消息也难以扩展，如果要添加另一参数（如加上HTTP方法），则必须重写整个消息。

第二个版本没有这些缺陷。由于参数列表具有清晰的语法，所以很容易被解析。而且它很容易阅读，为你能在参数前面看到完成的日志。它也很容易扩展，因为您只需将另一个参数添加到列表中即可。

区别WARNING和ERROR级别

显而易见，日志级别的存在是有原因的，你应该恰当的使用它们。

WARNING和 ERROR有一些关键的区别。

如果你执行了一些代码并产生了实际有效的结果，但是出现了一些问题，这时的日志是警告（WARNIG）。

如果你执行了一些代码但是没有达到预期的效果，这时就是一个错误（ERROR）。

让我们再看一个例子：

```
try {
    restClient.makeRequest()
    log.info("向REST API发送请求。[url={}]", url)
} catch(e: UnauthorizedException) {
    log.warn("请求REST API被拒绝，因为用户身份认证失败。 [url={}, result={}]", url, result)
} catch(e: Exception) {
    log.error("请求REST API失败。 [url={}, exception={}]", url, exception)
}
```

REST 调用可能具有以下三种结果之一：

- 正常执行，这是一个 INFO（在调用RestApi之后）。
- 因意外异常而失败。这是一个 ERROR（而不是INFO 日志）。
- 引发了一些预期的异常。这是一个 WARNING。

因此，在出现警告的情况下，你做了一些事情，但没有完美地完成。如果出现错误，你没有正常的做

件事情（出现也一个你没预料到的错误）。

另请注意，警告（当然也是错误）是号召性用语。如果没有人需要做出反应并做某事，那么您不需要录警告。

INFO记录业务相关的日志，DEBUG记录技术相关的日志

INFO 日志应该看起来像一本书。它应该告诉你发生了什么，而不一定是如何发生的。这意味着与技内容相比，INFO 更适合用于类似业务的日志。与技术相关的日志（通常）应该是 DEBUG。

```
INFO  | 发送用户注册邮件通知。[user="Thomas", email="thomas@tuhrig.de"]
INFO  | 邮件发送给用户 [user="Thomas"]
INFO  | 发送用户取消关注邮件通知。[user="Thomas", email="thomas@tuhrig.de"]
```

这种类型的日志从业务的角度告诉你一个故事。

现在什么是技术日志?

```
DEBUG | 将用户保存进通知列表。[user="Thomas", email="thomas@tuhrig.de"]
DEBUG | 发送欢迎邮件 [user="Thomas", email="thomas@tuhrig.de"]
INFO  | 发送用户注册通知 [user="Thomas", email="thomas@tuhrig.de"]
DEBUG | 开始发送当天邮件通知的定时任务。 [subscribers=24332]
INFO  | 邮件发送给用户 [user="Thomas"]
INFO  | 发送用户取消关注邮件通知。 [user="Thomas", email="thomas@tuhrig.de"]
```

每个（业务）用例都会生成一行 INFO 日志。此外，还有 DEBUG 日志可以更详细地了解该过程的工方式。

更多

当然，对于好的日志还有很多事情要做。您还需要考虑日志跟踪、日志聚合和指标等内容。但是说到粹的记录日志，我真的很推荐上面这些小规则。

致以我最好的问候

托马斯