融360日志规范

陈雕

chendiao@rong360.com

2016 年 11 月 23 日

第一节　概述

1.1　日志

日志是为了开发人员查看系统运行情况和追查系统bug而记录的文本. 遵循统一约定规范的, 结构化的日志记录能够以较小的存储成本, 帮助开发人员快速发现和定位系统问题.

日志不仅是面向系统开发人员的, 更是面向日志读者的. 比如如下日志记录:

```
ERROR: Save failed - SQLException ...
```

对于系统开发人员, 可能该日志已经足以说明问题, 开发人员也可以根据其中的信息找到并修复 bug; 但是对于非该系统的开发人员, 如下的日志记录能够更好的表述系统当前的问题:

```
ERROR: Save failed - Entity = Person, Data = {id = 123, data="I
am the data partion" } - SQLException ...
```

总的来说, 日志具有时间有序的特点, 其内容记录了系统中 **何时(WHEN, 时间戳) 何地(WHERE, 程序位置信息) 何事(WHAT, *Log body*)** 发生. 对于系统异常日志, 还需要给出事情发生的 **原因(WHY, *Exception stack trace*)**. 为了帮助日志分析人员重现问题, 则需要对事件发生 **过程(HOW)** 进行较为清晰的描述.

日志开源框架, 如 Log4J、Java Logging API、SLF4J 等, 均可通过简单配置, 在日志中打印 WHEN 和 WHERE 信息. 本规范目的是希望通过约

定规范日志中的 WHAT (包括 WHY 和 HOW) 信息, 并提供统一的、规范的日志接口, 使系统开发人员能够通过简单的配置和操作, 统一规范的记录日志.

第二节　规范

2.1　等级规范

根据日志记录内容对系统的影响情况, 本规范允许如下四个日志等级.

DEBUG 调试日志, 主要用于记录RD开发调试时的相关信息, 线上系统不记录该级别日志.

INFO 一般消息日志, 主要用于记录系统运行日志, 该级别日志属于系统正常运行时的日志

WARNNG 警告日志, 主要用于记录系统运行中的异常行为, 但是该异常行为不影响系统的正常运行

ERROR 错误日志, 主要用于记录系统运行中的错误, 这类错误可能会引起系统宕机

很多同学会在开发过程中使用 System.out.println 的形式记录日志, 以便调试. 但是该方式不能有效的系统记录的日志进行分级, 对日志的分析和理解造成不便, 在提交到代码库之前一定要记得删除这类测试代码. 一种备选方案是, 讲此类日志记录为 debug级别, 因为线上系统不会打印该级别日志, 因此不会对线上系统的日志分析造成困扰.

2.2　字段规范

为了方便定位系统问题, 记录的日志中需要包含如下信息:

`[日期 时间] [日志级别] [程序位置信息] [thread-name:currnt-time][logid] [ip] [uri] [merchant-id] [user-id] [session-id][current-time: ms] log-body`

日志各字段使用分隔符 **分隔符** 分离.

2.3　日志正文规范

为了方便查询问题, 日志正文格式如下：

[模块信息][流程信息][事件简单描述][事件输入参数(可选)][事件输出参数(可选)][异常堆栈(仅异常事件)]

以运营商抓取为例,

模块信息 指的是记录日志的模块或服务的名称, 其形式为: 北京移动*PC*端、上海移动*SHOP*端、中国移动*APP*端 等

流程信息 值得是记录日志的模块或服务的流程的名字, 其形式为: 登录、发送短信验证码、抓取通话详单等

事件简单描述 是开发人员对该日志事件的简单描述, 如: 登录失败, 服务密码错误、发送短信验证码失败 等

事件输入参数（可选） 是记录该日志的事件发生时, 提交给程序的参数, 如登录运营商系统的登录参数. 该字段为可选字段

事件输出参数(可选) 是记录该日志的事件发生时, 程序的输出信息, 如登录运营商系统失败时运营商的返回源码. 该字段为可选字段

异常堆栈(仅异常事件) 是异常事件发生时的异常堆栈. 该字段仅记录异常日志时打印

2.4　接口规范

2.4.1　@logConfig注解

当一个类需要使用日志规范时, 需要在定义类时添加 @logConfig 注解.该注解具有参数 module 参数, 该参数标明了该类所属模块.

2.4.2　@logFlow注解

当一个方法需要使用日志规范时, 需要在方法签名处添加 @logFlow 注解. 该注解具有一个参数 flow, 该参数标明了该方法所处的流程, 默认值为方法名.

2.4.3　在程序中记录日志

在程序中使用 `log.info()` 记录日志.

接口规范将在章节 三 中详细介绍.

第三节　实现

为了规范化日志, 本规范实现了统一的日志记录接口.

A　常见日志场景说明

- **关键函数入口日志**
  为了方便查看函数的调用情况, 可以在函数入口处打印函数入口日志, 记录为 INFO 级别日志, 需要打印出函数名和传入的实参的值, 如:

- **非关键函数入口日志**
  不建议记录非关键函数的入口日志, 如确因调试原因需要打印该类型日志, 记录为 DEBUG 级别日志, 需要打印出函数名和传入的实参值:

- **系统关键信息日志**
  以运营商抓取中的场景为例, 如判断短信验证码正确与否时, 短信验证码正确有唯一的识别标识(the-only-matched-condition), 则当验证码不匹配时, 记录级别为 WARN 的验证码不匹配日志, 其内容包括用户输入的验证码和运营商系统返回的源码:

  如果验证码正确没有较为明显的识别标识, 则可以明确表示验证码不匹配(the-not-matched-condition), 则不匹配情形记录 WARN 级别的验证码不匹配日志, 其内容包括用户输入的验证码和运营商系统返回的源码; 并对其它情形记录 INFO 级别的日志, 其内容包括运营商系统返回的源码:

- **系统异常日志**
  当系统执行捕获到异常时, 需要打印系统异常日志, 该类型日志记录为

WARN 级别, 需要打印出异常的具体调用栈信息, 如因数据问题引起的系统异常, 需要同时打印出引起异常的原始数据:

- **错误日志**
  当系统运行时需要无法修复的错误, 需要重启系统时, 需要记录错误日志, 日志级别为 ERROR, 需要打印出该错误的具体原因: