log applications

# Các framework hỗ trợ log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Framework** | **Supported log levels** | **Standard appenders** | **Popularity** | **Cost / licence** |
| [**Log4J**](https://en.wikipedia.org/wiki/Log4J) | FATAL ERROR WARN INFO DEBUG TRACE | AsyncAppender, JDBCAppender, JMSAppender, LF5Appender, NTEventLogAppender, NullAppender, SMTPAppender, SocketAppender, SocketHubAppender, SyslogAppender, TelnetAppender, WriterAppender | Widely used in many projects and platforms | Apache License, Version 2.0 |
| [**Java Logging API**](http://java.sun.com/javase/6/docs/technotes/guides/logging/) | SEVERE WARNING INFO CONFIG FINE FINER FINEST | Sun's default Java Virtual Machine (JVM) has the following: ConsoleHandler, FileHandler, SocketHandler, MemoryHandler |  | Comes with the JRE |
| [**Apache Commons Logging**](http://commons.apache.org/logging/) | FATAL ERROR WARN INFO DEBUG TRACE | Depends on the underlying framework | Widely used, in conjunction with log4j | Apache License, Version 2.0 |
| [**SLF4J**](https://en.wikipedia.org/wiki/SLF4J) | ERROR WARN INFO DEBUG TRACE | Depends on the underlying framework, which is pluggable |  | [MIT License](https://en.wikipedia.org/wiki/MIT_License) |
| [**tinylog**](http://www.tinylog.org/) | ERROR WARNING INFO DEBUG TRACE | ConsoleWriter, JdbcWriter, FileWriter, RollingFileWriter, SharedFileWriter and *null* (discards all log entries) [[1]](https://en.wikipedia.org/wiki/Java_logging_framework#cite_note-1) |  | Apache License, Version 2.0 |
| [**Logback**](http://logback.qos.ch/) | ERROR WARN INFO DEBUG TRACE | Too many to list: see [Appender JavaDoc](http://logback.qos.ch/apidocs/ch/qos/logback/core/Appender.html) | Used in many projects like [Akka](https://en.wikipedia.org/wiki/Akka_(toolkit)), [Apache Camel](https://en.wikipedia.org/wiki/Apache_Camel), [Apache Cocoon](https://en.wikipedia.org/wiki/Apache_Cocoon), [Artifactory](https://en.wikipedia.org/w/index.php?title=Artifactory&action=edit&redlink=1), [Gradle](https://en.wikipedia.org/wiki/Gradle),[Lift Framework](https://en.wikipedia.org/wiki/Lift_(web_framework)), [Play Framework](https://en.wikipedia.org/wiki/Play_Framework), [Scalatra](https://en.wikipedia.org/wiki/Scalatra),[SonarQube](https://en.wikipedia.org/wiki/SonarQube), etc... | [LGPL](https://en.wikipedia.org/wiki/LGPL), Version 2.1 |

## Simple Logging Façade for Java (SLF4J)

Simple Logging Façade for Java (SLF4J) như một framework abtract cho nhiều Logging framework khác (như java.util.logging, logback, log4j) cho phép người dùng có thể đưa vào logging framework mà họ mong muốn tại thời điểm triển khai.

SLF4J được tạo ra bởi Ceki Gülcü như một bản thay thế đáng tín cậy hơn của Jakarta Commons Logging framework. Một cuộc khảo sát vào năm 2013 trên 10,000 GitHub projects cho thấy rằng thư viên Java được dùng nhiều nhất là slf4j, với 30.7% dự án sử dụng nó.

Những dự án có sử dụng SLF4J:

|  |  |  |
| --- | --- | --- |
| * [Apache ActiveMQ](http://activemq.apache.org/) * [Apache Archiva](http://arhiva.apache.org/) * [Apache Camel](http://camel.apache.org/) * [Apache Directory](http://directory.apache.org/) * [Apache FTPServer](http://mina.apache.org/ftpserver/) * [Apache Geronimo](http://geronimo.apache.org/) * [Apache Graffito](http://incubator.apache.org/graffito/) * [Apache Jackrabbit](http://jackrabbit.apache.org/) * [Apache Mina](http://mina.apache.org/) * [Apache Qpid](http://cwiki.apache.org/qpid/) * [Apache ServiceMix](http://servicemix.apache.org/) | * [Apache Sling](http://incubator.apache.org/sling/site/index.html) * [Apache Solr](http://lucene.apache.org/solr/) * [Apache Tapestry](http://tapestry.apache.org/) * [Apache Wicket](http://incubator.apache.org/wicket/) * [Aperture](http://aperture.sourceforge.net/) * [Apogee](http://apogee.nuxeo.org/) * [Artifactory](http://www.jfrog.org/sites/artifactory/latest/) * [AsyncWeb](http://docs.safehaus.org/display/ASYNCWEB/Home) * [Bitronix](http://www.bitronix.be/) * [DbUnit](http://www.dbunit.org/) * [Display tag](http://displaytag.sourceforge.net/11/) | * [Ehcache](http://ehcache.org/) * [GMaven](http://groovy.codehaus.org/GMaven) * [Gradle](http://www.gradle.org/) * [GreenMail](http://www.icegreen.com/greenmail/) * [GumTree](http://gumtree.codehaus.org/) * [H2 Database](http://www.h2database.com/) * [HA-JDBC](http://ha-jdbc.sourceforge.net/) * [Hibernate](http://www.hibernate.org/) * [Igenko](http://code.google.com/p/igenko/) * [Jabsorb](http://jabsorb.org/) * [Jetty v6](http://jetty.mortbay.org/) |
| * [jLynx](http://www.topmind.biz/html/index.php) * [JMesa](http://code.google.com/p/jmesa/) * [JODConverter](http://www.artofsolving.com/opensource/jodconverter) * [JTrac](http://jtrac.info/dependencies.html) * [JWebUnit 2.x](http://jwebunit.sourceforge.net/2.x/) * [JQuantLib](http://www.jquantlib.org/index.php/Main_Page) * [LIFERAY](http://www.liferay.com/web/guest/home) * [Lift](http://liftweb.net/) * [log4jdbc](http://log4jdbc.sourceforge.net/) * [Magnolia](http://www.magnolia.info/en/magnolia.html) * [MRCP4J](http://mrcp4j.sourceforge.net/) | * [Mindquarry](http://www.mindquarry.com/) * [Mugshot](http://mugshot.org/) * [Mule](http://mule.codehaus.org/display/MULE/Home) * [Nexus](http://nexus.sonatype.org/repository.html) * [Novocode](http://www.novocode.com/naf/) * [NetCDF](http://www.unidata.ucar.edu/software/netcdf-java/) * [OpenMeetings](http://code.google.com/p/openmeetings/) * [OpenRDF](http://www.openrdf.org/) * [Penrose](http://docs.safehaus.org/display/PENROSE/Home) * [PZFileReader](http://pzfilereader.sourceforge.net/) * [Quartz Scheduler](http://www.quartz-scheduler.org/) | * [QuickFIX/J](http://www.quickfixj.org/) * [Sonar](http://sonar.codehaus.org/) * [SMSJ](http://smsj.sourceforge.net/dependencies.html) * [Spring-OSGi](http://www.springframework.org/osgi) * [SpringSource dm Server™](http://static.springsource.org/s2-dmserver/2.0.x/user-guide/htmlsingle/user-guide.html) * [StreamBase](http://streambase.com/) * [TimeFinder](http://www.timefinder.de/) * [WTFIGO](http://www.wtfigo.org/index.html) * [YASL](http://yaslibrary.sourceforge.net/index.shtml) * [Xooctory](http://xooctory.xoocode.org/) * [XWiki](http://xwiki.org/) |

# Các ứng dụng hỗ trợ log

Android Studio, và hầu hết tất cả các ứng dụng IDE hỗ trợ lập trình chủ yếu bằng Java.

Hệ thống log của Android cung cấp một cơ chế để thu thập và xem các thông tin debug được in ra. Logcat là một kho log lại những message của hệ thống, trong đó có cả những lỗi từ máy ảo cũng như từ chính ứng dụng bạn viết thông qua lớp Log.

## Lớp Log

Log là một lớp mà bạn có thể dùng trong code của mình để in message ra LogCat. Một số phương thức log phổ biến là:

* [v(String, String)](https://developer.android.com/reference/android/util/Log.html#v(java.lang.String, java.lang.String)) (verbose)
* [d(String, String)](https://developer.android.com/reference/android/util/Log.html#d(java.lang.String, java.lang.String)) (debug)
* [i(String, String)](https://developer.android.com/reference/android/util/Log.html#i(java.lang.String, java.lang.String)) (information)
* [w(String, String)](https://developer.android.com/reference/android/util/Log.html#w(java.lang.String, java.lang.String)) (warning)
* [e(String, String)](https://developer.android.com/reference/android/util/Log.html#e(java.lang.String, java.lang.String)) (error)

Ví dụ:

Log.i("MyActivity", "MyClass.getView() — get item number " + position);

Logcat sẽ in ra như sau:

I/MyActivity( 1557): MyClass.getView() — get item number 1

## Sử dụng LogCat

Bạn có thể sử dụng LogCat trong DDMS hoặc gọi bằng ADB shell. Để chạy LogCat, thông qua ADB shell, cú pháp tổng quát là:

[adb] logcat [<option>] ... [<filter-spec>] ...

Bạn có thể dùng lệnh LogCat từ máy tính phát triển ứng dụng hoặc thông qua adb shell từ xa trong máy ảo hoặc thiết bị. để xem log từ máy tính thì bạn dùng

$ adb logcat

và từ một adb shell từ xa bạn dùng

# logcat

Bảng sau mô tả các lựa chọn command line của logcat:

-c Xóa toàn bộ log và thoát.

-d Quăng log lên màn hình và thoát.

-f<filename> Viết log message xuống file. Mặc định là stdout.

-g In kích thước của log cụ thể và thoát

-n <count> Cài đặt số lần tối đa xoay vòng logs. Mặc định là 4. Yêu cầu phải có -r

-r <kbytes> Xoay vòng tập tin log mỗi <kbytes> được in ra. Mặc định là 16. Yêu cầu -f

-s Cài đặt mức lọc xuống silent.

-v <format> Cài đặt cấu trúc in ra cho log message. Mặc định là cấu trúc brief.

## Lọc Log

Mỗi log message trong android đều có một cái tag và một mức priority xác định.

* Tag là một chuỗi ngắn chỉ ra thành phần của hệ thống mà message đã được tạo ra.
* Priority là một trong những ký tự sau, sắp xếp từ thấp đến cao:
  + V — Verbose (mức thấp nhất)
  + D — Debug
  + I — Info
  + W — Warning
  + E — Error
  + F — Fatal
  + S — Silent (mức cao nhất, sẽ không có gì được in ra)

Bạn có thể lấy ra tag và priorities bằng cách chạy LogCat và lấy ra 2 cột đầu tiên trong mỗi message, tên là <priority>/<tag>.

Đây là một ví dụ tin nhắn có priority là I và tag là “ActivityManager”:

I/ActivityManager(  585): Starting activity: Intent { action=android.intent.action...}

Để giảm log ghi ra đến một mức quản lý được, bạn có thể dùng filter expression. Filter expression cho phép bạn chỉ ra cho hệ thống biết cặp tag và priority bạn quan tâm.

Filter expression có cú pháp như sau tag:priority …, tag chỉ ra tag mà bạn quan tâm, còn priority chỉ ra mức *thấp nhất* của tag đó. Những message nào thuộc tag đó mà có mức priority bằng hoặc cao hơn mức bạn đã định sẽ được ghi ra. Bạn có thể tạo ra bao nhiêu cú pháp như trên trong một filter expression cũng được. Chỉ cần cách nhau bởi dấu cách.

Dưới đây là một ví dụ loại bỏ tất cả log message trừ những message có tag là “ActivityManager”, ở mức “info” trở lên, tất cả log messages có tag là “MyApp” với priority “Debug” trở lên:

adb logcat ActivityManager:I MyApp:D \*:S

cái cuối cùng trong đoạn lệnh trên ,\*:S, đặt priority của tất cả các tag về “silent”, để đảm bảo rằng chỉ có log message với tag “ActivityManager” và “MyApp” được in ra.

Nếu bạn chạy LogCat trên máy tính thì ban có thể cài đặt filter expression mặc định bằng cách xuất giá trị cho biến ANDROID\_LOG\_TAGS:

export ANDROID\_LOG\_TAGS="ActivityManager:I MyApp:D \*:S"

## Kiểm soát định dạng log

Log message chứa một số trường metadata nhất định, ngoài tag và priority. Bạn có thể chỉnh định dạng xuất ra cho một message để chúng có thể hiện một trường metadata cụ thể nào đó. Để làm được điều đó, bạn dùng -v và đi kèm với một trong những định dạng trong danh sách dưới đây.

* brief - Hiện priority/tag và PID của quá trình message liên quan (định dạng mặc định)
* process - Chỉ hiện PID
* tag - Chỉ hiện tag và priority
* raw - Hiện log message chưa định dạng
* time - Hiện ngày, thời gian, priority, tag và PID, TID của tiểu trình message có liên quan.
* long - Hiện tất cả các trường metadata và cách các messsages bằng một dòng trống.

Khi bắt đầu LogCat, bạn có thể cụ thể một định dạng mà bạn muốn bằng cách dùng -v:

[adb] logcat [-v <format>]

Dưới đây là một ví dụ cho thấy cách tạo định dạng cho message trong thread:

adb logcat -v thread

Chú ý là khi dùng -v thì bạn chỉ có thể chỉ ra một định dạng mà thôi

## Xem stdout và stderr

Mặc định, hệ thống Android gửi đầu ra stdout và stderr (System.out và System.err) ra /dev/null. Trong các tiến trình chạy máy ảo Dalvik, bạn có thể nhờ hệ thống viết ra một bản copy của đầu ra xuống log file. Trong trường hợp này, hệ thống viết message xuống log bằng cách dùng log tags là stdout và stderr, cả hai với priority là I.

Để định hướng cho đầu ra bằng cách này, bạn hãy dừng máy ảo hoặc thiết bị và sau đó dùng shell command là setprop để mở việc điều hướng cho đầu ra. Đây là cách bạn làm điều đó:

$ adb shell stop  
$ adb shell setprop log.redirect-stdio true  
$ adb shell start

Hệ thống sẽ lưu cái cài đặt này cho đến khi bạn tắt máy ảo hoặc thiết bị. Để dùng cài đặt này làm mặc định trên máy ảo hoặc thiết bị, bạn có thể thêm một entry vào /data/local.prop trên thiết bị.