

App Note: AylaLog in Android

This document describes how logs are created and saved in the Ayla SDK, and describes how to use the SDK class AylaLog to manage logs in the app. Ayla mobile SDK uses the AylaLog class to write logs to the console, and to store them in log files, which can then be sent to mobile-support via email from within Aura and AMAP apps.

1. Initialize AylaLog

Initialize AylaLog using the method `AylaLog.initAylaLog(String fileName, LogLevel consoleLevel, LogLevel fileLevel)` after initializing the Ayla library. This method takes the preferred log file name, console log level, and file log level as parameters.

eg: `AylaLog.initAylaLog("ayla_logs", LogLevel.Debug, LogLevel.Error);`

The SDK supports six log levels defined in the enumeration `AylaLog.LogLevel` - Verbose, Debug, Info, Warning, Error and None.

The log levels can be modified using methods `setConsoleLogLevel()` and `setFileLogLevel()`. Make sure that no sensitive information is written to log files.

2. Writing logs to file and console

Once the log levels are set, logs can be written to both console and file using the methods `AylaLog.d(TAG, message)`, `AylaLog.e(TAG,message)` etc. TAG is a String used to identify the source of the log message. It is a good practice to use a unique TAG for each class. The logs written using these methods will be filtered according to the set log levels before writing to the file or console. For example, setting log level 'Debug' will include logs with levels Debug, Info, Warning and Error, and setting log level 'Info' will include logs with levels Info, Warning and Error.

Eg: Writing a console log using the statement `AylaLog.d("LanModule", "Lan command queue is empty")` will create the following log in Android Studio console.

```
"com.aylanetworks.aura D/LanModule: Lan command queue is empty"
```

Here, the logs starts with package name of the app from which it was written to the console. The letter "D" indicates that it is a debug log. An info log will be denoted by "I", error log by "E", warning log by "W", and a verbose log by "V". The text following the log level is the TAG parameter passed to the method. Here the TAG is "LanModule". The text following the TAG is the log message passed to the method. Here the message parameter is "Lan command queue is empty".

3. Overriding set log levels

AylaLog class also includes methods to write logs regardless of the set log levels.

`consoleLogDebug(String TAG, String message)` creates a debug log with the set TAG and message regardless of the log levels set in the app. Similarly `consoleLogError(String TAG, String message)` creates an error log regardless of the set log level.

4. Email logs

To email the logs saved on the phone, apps should get an Intent object from the SDK using `getEmailIntent(Context context, String[] supportEmail, String emailSubject, String emailMessage)`. This method returns an Intent object that can be used to open a preferred email app on the phone. Email subject and text passed to the method will be populated in the email draft and the two most recent log files will be attached in the email draft.