

0. 日志能力建设

- 1.背景
- 2.目标
- 3.实施方案
 - 3.1 日志主动上报
 - 3.2 日志回捞能力建设
 - 3.3 日志工具
- 4.日志规范制度
 - 4.1 开发规范
 - 4.2 审核规范
 - 4.3 操作 SOP
- 5.案例分析

1.背景

SSC Android 当前在日志使用方面还不是很规范，正常、异常的链路日志都比较缺失，需要约束和规范下来。同时，在用户遇到问题的时候，无法获取到 App 的日志，无法用客户端日志来协助排查问题，需要加强。

2.目标

1. 建立 App 主动上报、被动回捞的能力，确保日志获取成功率达到 100%
2. 完善客户端日志工具
3. 建立日志的规范制度
 - a. 事前：日志的要求
 - b. 事中：日志的监督
 - c. 事后：日志操作 SOP 流程

3.实施方案

3.1 日志主动上报

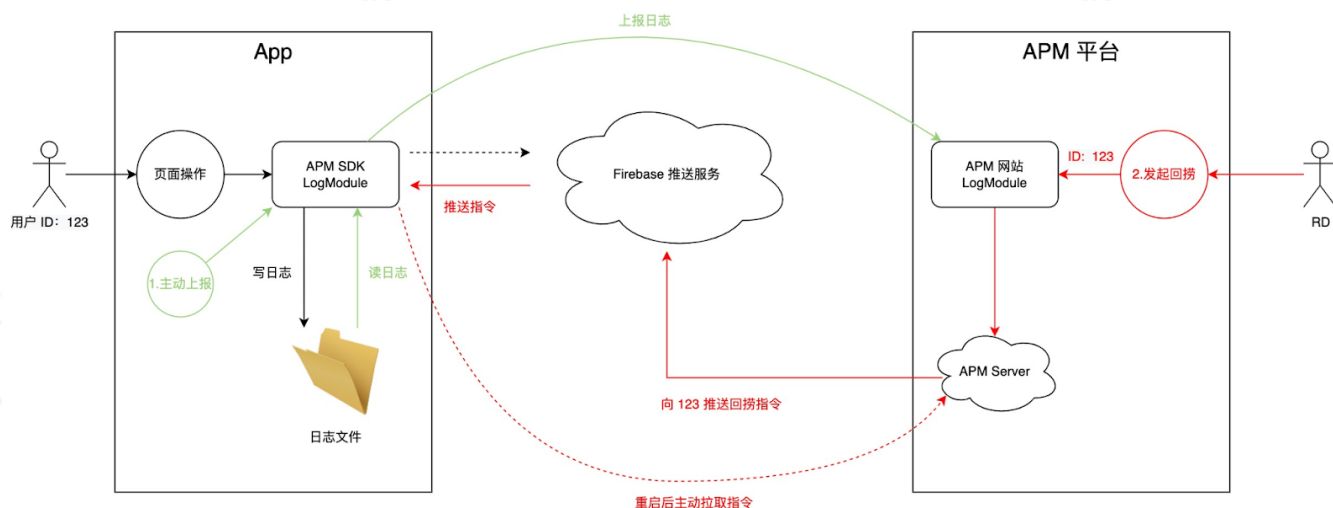
用户端日志因为数据量比较大，一般是存储在用户本地，一旦遇到问题，需要用到时才去做上传。

- App 预留主动上传日志的机制，如连续点击多次 ICON 上传
- 具体上传流程可以通过用户侧 SOP 来指引

业内常规做法：关于页 → 版本信息 → App 图标上连续点击 5 次上传，成功后 Toast 提示

3.2 日志回捞能力建设

利用 MDAP 回捞的能力，在联系不上用户的时候可将用户手机上的日志回捞到 MDAP 平台上。在用户登录账号的情况下，使其达到



3.3 日志工具

日志工具

4.日志规范制度

4.1 开发规范

日志分级（和 Android 原生保持一致）

- verbose
- debug：调试诊断信息
- info：记录系统正常运行的信息，如方法名、参数、服务端请求结果关键信息等
- warn：运行时出现错误，但是还可以正常运行
- error：运行出现错误，且影响了业务流程

日志要求

- 线上关闭 debug 日志，只保留 info 级别的日志
- 重要流程：info 记录调用轨迹
- 异常分支：
 - 不影响业务：warn
 - 影响业务：error

4.2 审核规范

Code Review 代码的时候需要对日志规范做要求，不符合的需要修改。

4.3 操作 SOP

- RD：SOP-RD
- 用户：SOP-用户

5.案例分析

1. [ServicePoint] inbound - self collection 页面扫码慢的问题排查