# 异常日志系统

1. **简介**

作为第三方平台，sdk的稳定性以及异常错误的定位和解决是用户参考的重要指标，因此除了提供用户所需的功能外。我们也需要建立一套自己的异常反馈系统。

1. **log日志格式**

log日志文件保存在手机sdcard中，以天为单位，文件命名为PASDK\_20170314，日志格式为时间 ｜日志级别 ｜线程id ｜日志标签（Tag）｜当前类文件＋行号＋方法名｜具体的log日志

例如：

2016/11/17 19:07:59.187 ｜ E ｜3434｜PASDK ｜AVRoom.java(100):Create |

udp server 163.212.21.205:8000

其中，日志级别包括:

D(Debug): 包含了非常广泛的上下文信息，用于问题诊断。

I(Info): 用于在产品环境中（粒度较粗）帮助跟踪执行过程的上下文消息。

W(Warning): 警告消息，说明系统中可能存在问题。例如，如果这个消息类别是有关安全性方面的。

E(Error): 错误消息说明系统中出现了严重的问题。这种问题通常都是不可恢复的，需要人工进行干预。

日志标签Tag根据模块进行统一，这样方便后面的日志筛选。比如：

IM模块 （PA\_IM\_\*\*）

推流模块 （PA\_PUSH\_\*\*）

业务服务器模块 (PA\_BSER\_\*\*)

1. **log日志包括的内容**

l错误定位需要的信息： 运行环境，第三方与sdk的交互记录。 sdk内部各个模块之间的交互记录（比如，IM模块与 推流模块，业务服务器模块之间的交互）。以及推流或播放的流程记录

log日志记录内容包括以下几个部分：

1.运行环境,因为该环境基本不会改变,所以写入log日志文件的开头部分,格式：

手机系统 ＃系统版本号＃手机型号＃手机网络供应商（电信｜移动｜联通）

Android ＃5.0 ＃ HUAWEI P8 ＃chinatelecom

2.调用api记录，第三方app调用sdk api时，保存当前调用的时间和方法名和传入的参数值

比如sdk 初始化函数 : AVContext::init( context, testAppid);

3.sdk回调第三方的接口记录。 保存sdk回调调用的时间和方法名和传入的参数值

比如sdk 切换摄像头回调 : AVContext::onComplete( 0, 1);

4.sdk与业务服务器接口调用记录。保存sdk调用服务器的请求以及返回。这里是否要考虑安全保密的问题，决定是否要加入调用的参数，以及服务器返回的数据。

比如进入房间 AVContext::enterRoom( 3010,true );

AVContext::enterRoom callback( {ret:1});

5.推流状态的记录。从获取推流地址－－开始推流－－推流成功－－－推流结束的状态记录

比如 pushState#getPushUrl#url:\*\*\*\*

pushState#startPush

pushState#pushSuccess

pushState#pushFailed#reason

pushState#pushEnd

6.播放器状态的记录。从获取播放地址－－开始播放（多路播放）－－停止播放的记录

比如 playState#getPushUrl#url:\*\*\*\*|url:\*\*\*\*

playState#startPlay#url:\*\*\*\*|url:\*\*\*\*

playState#firstFrameGet#url:\*\*\*\*|url:\*\*\*\*

playState#endPlay#url:\*\*\*\*|url:\*\*\*\*

7.IM信令的记录。（这里是否要加密）

比如 收到主播推流的IM . IMCMD#cameraEvent|hostId