日志库：com.fj.fjxlog:fjxlog_construction:1.1.0.74-RELEASES时序图

集成

1. ## maven依赖

```
api 'com.fj.fjxlog:fjxlog_construction:1.1.0.74-RELEASES'
```

2. ## 环境激活

判断当前是否是主进程

```
/**
 * @param context
 * @param pid
 * @description:获取当前进程名
 */
public static String getAppName(Context context, int pid) {
    String processname = "";
    ActivityManager am = (ActivityManager)
context.getSystemService(Context.ACTIVITY_SERVICE);
    List<ActivityManager.RunningAppProcessInfo> runningAppProcesses =
am.getRunningAppProcesses();
    Iterator<ActivityManager.RunningAppProcessInfo> iterator =
runningAppProcesses.iterator();
    while (iterator.hasNext()) {
        ActivityManager.RunningAppProcessInfo next = iterator.next();
        try {
            if (next.pid == pid) {
                processname = next.processName;
                return processname;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return processname;
}
   /**

     * @description:初始化log
     * @return: void
     */
     public static void initLog(Application context, String sn) {
        String appName = getAppName(context, Process.myPid());
        Log.e("appName", appName);
        if (TextUtils.isEmpty(appName)) {
            return;
```

```
        }
        if (appName.contains(":remote")) {
            initRemoteLog(context);
        } else {
            initMainLog(context, sn);
        }

    }


    private static void initMainLog(Application application, String sn) {
             UpInfoBuilder upInfoBuilder = new UpInfoBuilder()
                .putSn(sn)//sn号
                .putMachineType(Constants.LOG_MACHINE_TYPE) //机器类型（1：挖掘机；  2：旋挖钻；  3：压路机）
                .putUserName(KVProvide.getInstance().getAccountName())//用户
                .putProjectName("pile")//项目名称
                .putLogVersion(BuildConfig.VERSION_NAME);//当前应用版本
     // XLogConstant.ENV.RELESE;国内正式
      //  XLogConstant.ENV.TEST;测试环境
      //  XLogConstant.ENV.FOREIGN 海外环境
        XLogHelper.Companion.getInstance().init(application, BuildConfig.DEBUG, "pile", sn, upInfoBuilder, XLogConstant.ENV.RELESE);
        //        context: Application?, 应用上下文
        //        isDebug: Boolean,//是否控制台输出
        //        name: String = "digger",//项目名
        //        sn: String = "testsn",//sn
        //        upInfoBuilder: UpInfoBuilder,//配置对象
        //        evn: XLogConstant.ENV? = null//环境枚举
    }
  /**
     * description 高频日志初始化，高频日志只提供写入高频目录文件的能力，上传还是通过主进程进行操作
     * author tony.tang
     * date 2021/11/26 14:42
     */
    private static void initRemoteLog(Application application) {
        XLogHelper.Companion.getInstance().init2(application, BuildConfig.DEBUG, "pile");

 XLogHelper.Companion.getInstance().setLogVersion(BuildConfig.VERSION_CODE + "");
    }
```

使用

写入日志库api 通过provide写到子进程中的xlog中

```
//debug 高频
FJXLog.INSTANCE.hd("ECU-Get#", DataUtil.byte2hex(data));
//error 高频
FJXLog.INSTANCE.he("ECU-Get#", DataUtil.byte2hex(data));
//warming 高频
```

```
FJXLog.INSTANCE.hw("ECU-Get#", DataUtil.byte2hex(data));
日常日志api

//debug 普通日志
FJXLog.INSTANCE.d("ECU-Get#", DataUtil.byte2hex(data));
//error 普通日志
FJXLog.INSTANCE.e("ECU-Get#", DataUtil.byte2hex(data));
//warming 普通日志
FJXLog.INSTANCE.w("ECU-Get#", DataUtil.byte2hex(data));
```

3. 上传日志 写入调用上传日志的activity或者fragment

```
   // 成功数量
    private int sucsize = 0;
    //总共上次数量
    private int totalsize = 0;
    //失败上传数量
    private int failsize = 0;

    /**
     * description 重置数据
     */
    public void reset() {
        sucsize = 0;
        failsize = 0;
        totalsize = 0;
    }

       /**
     * description 上传日志
     * @param isapp 普通日志 true，高频日志 false
     */
    public void upAppLog(boolean isapp) {
        if (!NetworkUtil.isNetworkAvailable(this)) {
            MyToast.error(R.string.network_disconnect);
            return;
        }
        showLoadingDialog(R.string.dialog_msg_uploading);
        //在这里重新设置用户信息的用户名

 XLogHelper.Companion.getInstance().setUserName(accountProvider.getAcount())
;
        XLogHelper.Companion.getInstance().startUploadFile(
                // HttpUtils.CONSTRUCTION_BASE_URL + "log/app/push", 0,
param2,
                new XLogUploadListener() {
                     /**
                     * 上传错误信息回调
                     * @param code 错误code
                     * @param msg 错误信息，  现在只要判断code，不是未初始化，就结束
dialog
```

```
                     */
                    @Override
                    public void error(int code, @NonNull String msg) {
                        FJXLog.INSTANCE.e("XLogHelpererro", "错误信息" + msg);
                        if (code != XLogConstant.CODE_ERROR_BUILD_NOINIT) {
                            dismissLoadingDialog();
                            FJXLog.INSTANCE.e("XLogHelpererro", msg);
                            MyToast.error(R.string.network_exception);
                            //上传过程中，弱网，断网，都会中止整个上传，
                            //重新点击，会直接跳过压缩文件，将原来的压缩文件上传cos
                        }
                    }

                    /**
                     * description 上传完成
                     */
                    @Override
                    public void upover() {
                        MyToast.successL(String.format(Locale.getDefault(),
getString(R.string.toast_upload_log_finish), sucsize, totalsize
                                , failsize, totalsize));
                        dismissLoadingDialog();
                        reset();
                    }

                    /**
                     * description 上传数量改变的回调
                     */
                    @Override
                    public void onUploadStatus(boolean noFile, boolean
success, @org.jetbrains.annotations.Nullable String fileName, int allSize,
int surplus,
                                               int successSize, int
errorSize) {
                        try {
                            if (noFile || allSize == 0) {

 MyToast.info(getString(R.string.error_no_log_files));
                                dismissLoadingDialog();
                            } else {
                                if (surplus == 0) {
                                    if (errorSize == 0) {
                                        sucsize = successSize;
                                        totalsize = allSize;
                                        failsize = errorSize;
//
MyToast.successL(String.format(Locale.getDefault(),
getString(R.string.toast_upload_log_finish), successSize, allSize
//                                                  , errorSize, allSize));
                                    } else if (errorSize == allSize) {

 MyToast.errorL(String.format(Locale.getDefault(),
getString(R.string.toast_upload_log_finish), successSize, allSize,
                                                   errorSize, allSize));
                                    } else {
```

```
 MyToast.warnL(String.format(Locale.getDefault(),
getString(R.string.toast_upload_log_finish), successSize, allSize,
                                                errorSize, allSize));
                                    }
                                } else {
                                    int percent = 100 - 100 * surplus /
allSize;

 showLoadingDialog(String.format(Locale.getDefault(),
getString(R.string.dialog_msg_upload_progress), percent));
                                }
                            }
                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                    }
                }
                , isapp);
    }
```

4. 注意事项 如果有第三方库用mmap方式实现了文件写入，如mmkv，请在android{ defaultConfig{ *}}内将一下代码添加到build.gralde中

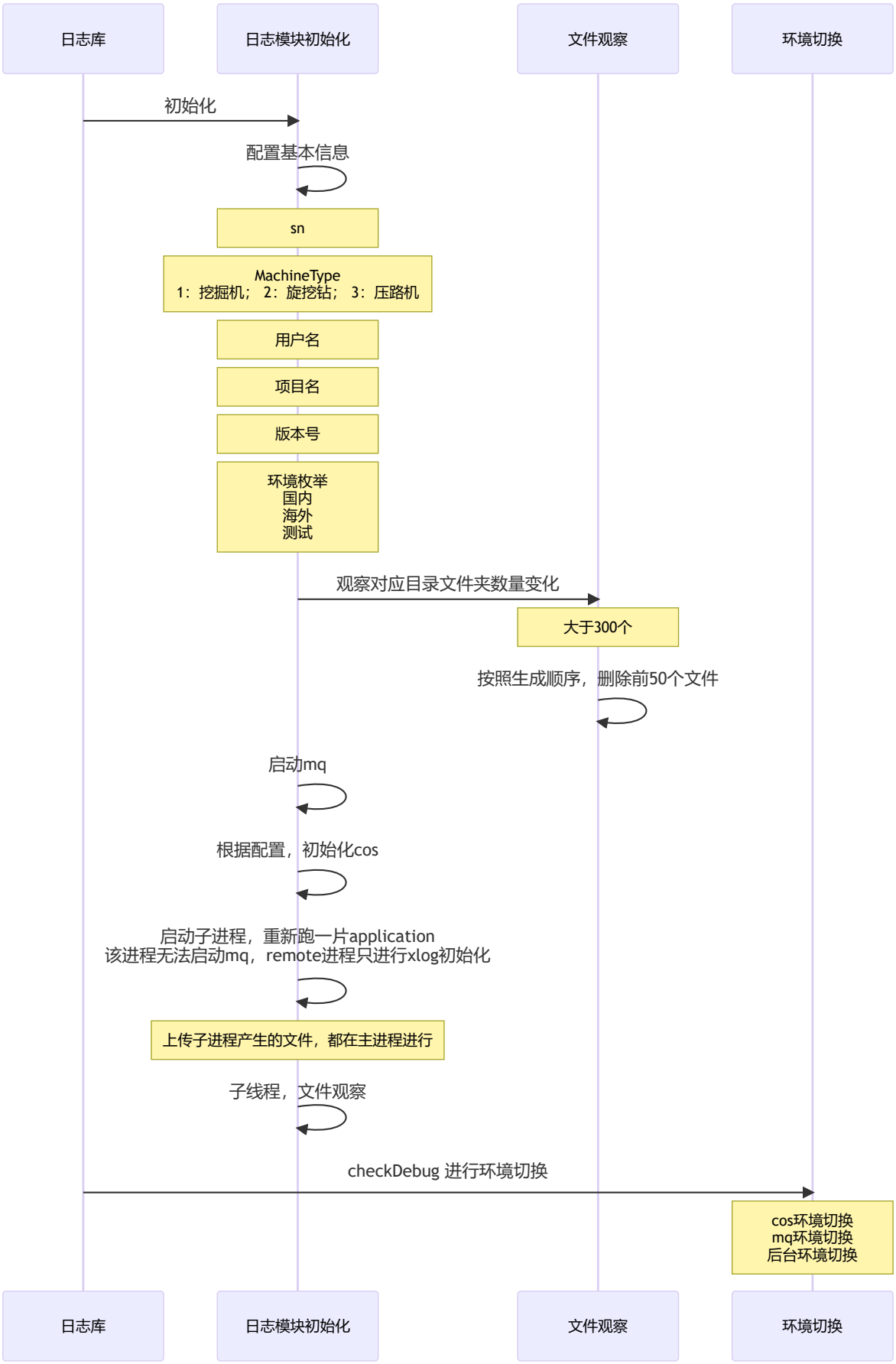
```
defaultConfig{
     packagingOptions{
               pickFirst 'lib/arm64-v8a/libc++_shared.so'
               pickFirst 'lib/x86/libc++_shared.so'
               pickFirst 'lib/armeabi-v7a/libc++_shared.so'
          }
}
```

5. 主工程manifest

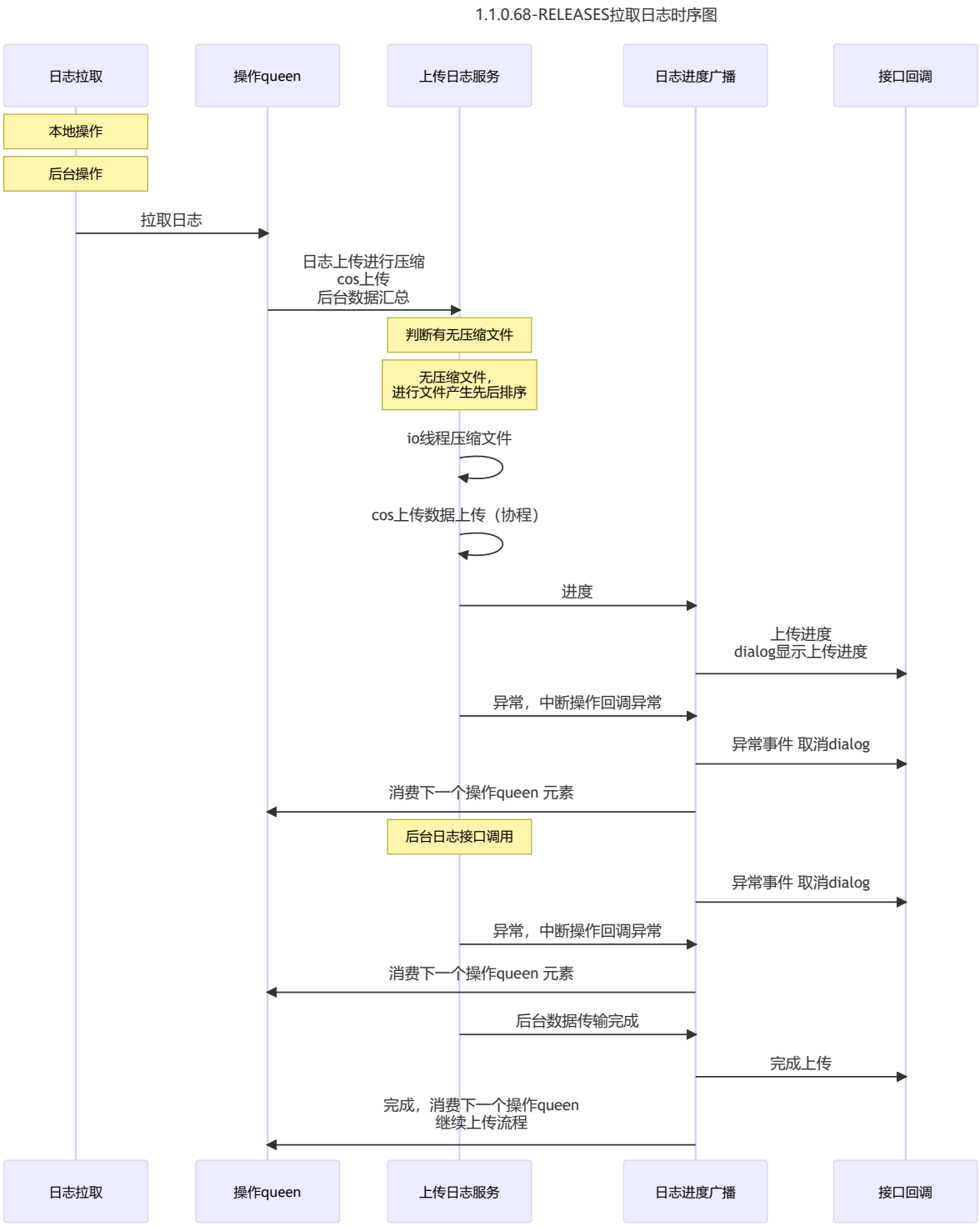
```
<service
           android:name="org.eclipse.paho.android.service.MqttService"
           android:enabled="true"
           android:exported="true" />
```

日志初始化时序

1.1.0.68-RELEASES拉取日志时序图


日志库
日志模块初始化
文件观察
环境切换
初始化
配置基本信息
sn
MachineType
1：挖掘机； 2：旋挖钻； 3：压路机
用户名
项目名
版本号
环境枚举
国内
海外
测试
观察对应目录文件夹数量变化
大于300个
按照生成顺序，删除前50个文件
启动mq
根据配置，初始化cos
启动子进程，重新跑一片application
该进程无法启动mq，remote进程只进行xlog初始化
上传子进程产生的文件，都在主进程进行
子线程，文件观察
checkDebug 进行环境切换
cos环境切换
mq环境切换
后台环境切换
日志库
日志模块初始化
文件观察
环境切换

日志拉取操作时序


1.1.0.68-RELEASES拉取日志时序图
日志拉取
操作queen
上传日志服务
日志进度广播
接口回调
本地操作
后台操作
拉取日志
日志上传进行压缩
cos上传
后台数据汇总
判断有无压缩文件
无压缩文件，
进行文件产生先后排序
io线程压缩文件
cos上传数据上传（协程）
进度
上传进度
dialog显示上传进度
异常，中断操作回调异常
异常事件 取消dialog
消费下一个操作queen 元素
后台日志接口调用
异常事件 取消dialog
异常，中断操作回调异常
消费下一个操作queen 元素
后台数据传输完成
完成上传
完成，消费下一个操作queen
继续上传流程
日志拉取
操作queen
上传日志服务
日志进度广播
接口回调

存储规则

sdcard 存储目录

> "sdcard/FJ/log/${项目名}/applog" 普通日志缓存地址
> "sdcard/FJ/log/${项目名}/hightwritelog" 高频日志缓存地址

压缩生成的缓存日志

> "sdcard/FJ/log/${项目名}/applog/temp${临时文件序号}.gz" 普通日志压缩后缓存地址
>
> 上传成功，后台接口调用成功后，会将gz文件删除掉

上传文件cos的命名规则

> ```
> var zipCosKey =
> 有项目名 "/项目名" 无 ""
> 有用户名 "/用户" 无 ""
> 有sn "/sn" 无 ""
> 有是否普通日志 "/applog" 否 "hightwritelog"
> 文件名 Fj_"yyyyMMddHHmmss_毫秒数.gz"
>
> 例如
> /pile/15814886147/applog/Fj_20220222101759_1645496279563.gz
> ```

faq

1. android高版本不支持mq远程推送
2. 文件存储不支持28以上版本

测试情况

75		8	上传日志	点击进入设置-账户信息	1、点击上传日志 2、无网络条件下点击上传日志 3、弱网条件下上传日志（限速为1Kb/s） 4、上传日志途中断开网络，连接网络再重新上传日志	1、上传日志成功 2、提示连接网络 3、上传成功，或失败提示网络异常，程序不会崩溃 4、断开网络提示网络异常，连接网络后重新上传成功			
76		9	上传高频日志	点击进入设置-账户信息	1、点击上传高频日志 2、无网络条件下点击上传日志 3、弱网条件下上传日志（限速为1Kb/s） 4、上传日志途中断开网络，连接网络再重新上传日志	1、上传日志成功 2、提示连接网络 3、上传成功，或失败提示网络异常，程序不会崩溃 4、断开网络提示网络异常，连接网络后重新上传成功			

各个版本情况说明

1.1.0.74-RELEASES

新增秘钥
公钥
val
PUB_KEY="84e9ba36ef2f1e912fd472c2924edb707617788dbe561e15a7564a6cbd08b7e6637d5eb9fa86b1f1b707062336c8391ae972afab36478c9766636da6e2c9be69"

公钥 私钥 解密有用（下为私钥）
85472b071be23389aa87037b3e2fabc2ab6bcb67c7bee3bf5bf7d0decf7987f

1.1.0.73-SNAPSHOT

每次xlog生成文件后

会将产生的xlog文件进行排序

如果新增是gz文件，则放弃后续操作

如果倒数第二个文件是gz文件，依然放弃操作

如果倒数第二个文件时.xlog 文件，将文件名改为FJ_时间戳 文件

1.1.0.68

com.fj.fjxlog:fjxlog_construction:1.1.0.68-RELEASES
解决混淆类冲突问题
第三方包会被打成a.a.a 现在混淆包打包成com.fjxlog

1.1.0.66

提供切换环境的api
XLogHelper.instance.checkDebug(false,XLogConstant.ENV.FOREIGN)//海外环境

1.1.0.59-RELEASES

上传日志api startUploadFile参数更改为接口+是否普通日志的布尔值