sdk 文件导入

使用 lua 接口前，要接入 sdk，详细的接入流程详见下载链接中的说明，
http://dev.applogger.cn/assets/download/logger_sdk.zip。

Lua 库文件添加

将 AppLogger_lua.h 和 AppLogger_lua.cpp 加入到项目中,随项目一起编译。
这两个文件是 AppLogger 的 lua 库封装实现。

Lua 接口说明

除了sdk初始化接口，AppLogger中的其他接口都做了lua的封装。
android和ios的sdk初始化接入，按sdk文档接入即可。

下面为lua接口的一些说明。

1. 初始化
   游戏 lua 环境加载完成以后，开始初始化统计接口 lua 环境，使用如下代码：：
   cocos2d-x 为例，在 AppDelegate.cpp 中添加:

```
#include "AppLogger.h"
#include "AppLogger_lua.h"

bool AppDelegate::applicationDidFinishLaunching()
{
    ……
    // pEngine 为游戏内第一次调用 LuaEngine::getInstance()的值
    AppLogger_lua::init(pEngine->getLuaStack()->getLuaState());
    ……
}
```

2. 引入
   在需要调用接口的 lua 文件中，加入下行:
   `require "AppLogger"`

3. 参数和返回值

AppLogger.setDebugLog(mode, level)	调试日志输出 model：输出模式，类型为number 0 // 不输出调试信息 1 // 输出到控制台（logcat/console） 2 // 输出到文件（applogger.log） 3 // 输出到控制台和文件 level：调试信息级别，类型为number 0 // 输出所有级别调试信息 1 // 输出INFO级别以上调试信息 2 // 输出WARN级别以上调试信息 3 // 输出ERROR级别以上调试信息
AppLogger.setSessionInterval (v)	设置 session 间隔 单位秒 v：间隔时间，类型为 number
AppLogger.enableCrashReport()	启用崩溃统计，适用于 java/c/c++/objective-c 代码
AppLogger.enableOnlineConfig()	开启在线参数
AppLogger.getOnlineConfig (param)	获取指定的在线参数 param：要获取的后台参数，类型为 string 此接口返回值为 string
AppLogger.onSubStart (stage)	进入关卡 stage：关卡名称，类型 string 和 number 均可 可转化为数字的 string 和 number 可在关卡中展示 其他 string 在页面分析中展示
AppLogger.onSubEnd (stage)	退出关卡 参数说明同上
AppLogger.onPassFail (bPass)	关卡成功或者失败 bPass：过关结果，类型为number 0 // 过关失败 1 // 过关成功
AppLogger.onEvent (name, stage)	自定义事件 name：事件名称，类型为 string stage：此事件所在的关卡，类型为 string 可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.onBuy	购买事件

(service,item,count, value, stage)	service：付款来源，类型为 string item：计费点，类型为 string count：计费点个数，类型为 number value：所付金额，类型为 number 或者 string stage：此事件所在的关卡，类型为 string 1.value 实际值带小数时，可使用 string 2.可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.onUse (item,count,stage)	道具使用 item：道具名称，类型为 string count：道具个数，类型为 number stage：此事件所在的关卡，类型为 string 可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.onExchange (item,count,stage)	游戏中的兑换 item：兑换的项目，类型为 string count：兑换数量，类型为 number stage：此事件所在的关卡，类型为 string 可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.onCollect (item,count,stage)	游戏中的产生的掉落和收集 item：收集的项目，类型为 string count：收集的个数，类型为 number stage：此事件所在的关卡，类型为 string 可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.onReward (item,count,stage)	游戏中的奖励 item：奖励的项目，类型为 string count：奖励的个数，类型为 number stage：此事件所在的关卡，类型为 string 可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.onShare (service,item,stage)	分享后的计数统计 service：已经分享到（微博，微信），类型为 string item：分享说明，类型为 string stage：此事件所在的关卡，类型为 string

	可以不使用 stage 参数，这种情况下默认为最后关卡
AppLogger.clearStatus()	状态值清理
AppLogger.setStatus (key,value,bAutoConvert)	设置状态值 key：状态关键字，类型为string value：状态值，类型为number bAutoConvert ：是否分区，类型为number，默认为 1 1 // 自动分区 0 // 固定分区
AppLogger.setUser (level,age,gender,userId,service)	用户统计 level：等级，类型为 string age：年龄，类型为 number gender：性别，类型为 string userId：唯一标识(邮箱等)，类型为 string service：所在区服，类型为 string 除 level 必须指明外，剩余 4 个参数都有默认值，调用时可以不使用