Bugly iOS SDK 2.0 使用文档

1. SDK 集成

- 下载并解压 SDK
- 拖拽 `Bugly.framework` 文件到 Xcode 工程内 （请勾选 `Copy items if needed` 选项）
- 添加依赖库
  - `SystemConfiguration.framework`
  - `Security.framework`
  - `libz.dylib`

2. 初始化 SDK

1.导入头文件

在工程的 `AppDelegate.m` 中导入头文件

`#import <Bugly/Bugly.h>`

如果是 Swift 工程，请在对应 `bridging-header.h` 中导入

2.初始化 Bugly

在工程 `AppDelegate.m` 的 `application didFinishLaunch...` 方法中初始化 Bugly

Objective-C

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary
    [Bugly startWithAppId:@"此处替换为你的AppId"];
    return YES;
}
```

Swift

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [
    Bugly.startWithAppId("此处替换为你的AppId")
    return true
}
```

至此，恭喜你的工程已经成功集成 Bugly，接下来编译并运行你的工程吧：）

3. 配置项

Bugly 支持通过 `Info.plist` 配置SDK相关参数

```
- Appid
    - Key: BuglyAppIDString
    - Value: 字符串类型
- 渠道标识
    - Key: BuglyAppChannelString
    - Value: 字符串类型
- 版本信息
    - Key: BuglyAppVersionString
    - Value: 字符串类型
- 开启Debug信息显示
    - Key: BuglyDebugEnable
    - Value: BOOL类型
```

在工程的 `Info.plist` 内添加对应的key及value配置SDK相关参数

4. 自定义日志功能

Bugly 提供自定义日志打印接口，用于记录一些关键的业务调试信息，可以更全面地反应App发生崩溃或异常时的上下文环境。使用方式与NSLog一致，可控制日志打印级别，是否在Console显示等，接口如下：

```
BLogError(fmt, …)
BLogWarn(fmt, …)
BLogInfo(fmt, …)
BLogDebug(fmt, …)
BLogVerbose(fmt, …)
```

在调用日志打印前，须先调用初始化方法启动日志打印模块

```
//设置日志打印级别为BLYLogLevelWarn且日志不打印到控制台
[BuglyLog initLogger:BLYLogLevelWarn consolePrint:NO];
```

5. SDK接口及设置

1. BuglyConfig

```objc
/**
 *  SDK Debug 信息开关，默认关闭
 */
@property (nonatomic, assign) BOOL debugMode;
/**
 *  设置自定义渠道标识
 */
@property (nonatomic, copy) NSString *channel;
/**
 *  设置自定义版本号
 */
@property (nonatomic, copy) NSString *version;

/**
 *  设置自定义设备唯一标识
 */
@property (nonatomic, copy) NSString *deviceId;
/**
 *  卡顿监控开关，默认关闭
 */
@property (nonatomic, assign) BOOL blockMonitorEnable;
/**
 *  卡顿监控判断间隔，单位为秒
 */
@property (nonatomic, assign) NSTimeInterval blockMonitorTimeout;
/**
 *  ATS开关，默认开启
 */
@property (nonatomic) BOOL appTransportSecurityEnable;
/**
 *  进程内还原开关，默认开启
 */
@property (nonatomic) BOOL symbolicateInProcessEnable;
/**
 *  非正常退出事件记录开关，默认关闭
 */
@property (nonatomic) BOOL unexpectedTerminatingDetectionEnable;
/**
 *  页面信息记录开关，默认开启
 */
@property (nonatomic) BOOL viewControllerTrackingEnable;
/**
 *  Bugly Delegate
 */
@property (nonatomic, assign) id<BuglyDelegate> delegate;
```

1.1 Bugly Delegate 异常回调

```
/**
 *  发生异常时回调
 *
 *  @param exception 异常信息
 *
 *  @return 返回需上报记录，随异常上报一起上报
 */
- (NSString *)attachmentForException:(NSException *)exception;
```

2. Bugly 接口

```
/**
 *  初始化Bugly,使用默认BuglyConfig
 *
 *  @param appId 注册Bugly分配的应用唯一标识
 */
+ (void)startWithAppId:(nullable NSString *)appId;

/**
 *  使用指定配置初始化Bugly
 *
 *  @param appId 注册Bugly分配的应用唯一标识
 *  @param config 传入配置的 BuglyConfig
 */
+ (void)startWithAppId:(nullable NSString *)appId
                config:(nullable BuglyConfig *)config;

/**
 *  设置用户标识
 *
 *  @param userId 用户标识
 */
+ (void)setUserIdentifier:(nonnull NSString *)userId;

/**
 *  设置关键数据，随崩溃信息上报
 *
 *  @param value
 *  @param key
 */
+ (void)setUserValue:(nonnull NSString *)value
              forKey:(nonnull NSString *)key;

/**
```

```
 *  获取关键数据
 *
 *  @return 关键数据
 */
+ (nullable NSDictionary *)allUserValues;

/**
 *  设置标签
 *
 *  @param tag 标签ID, 可在网站生成
 */
+ (void)setTag:(NSUInteger)tag;

/**
 *  获取当前设置标签
 *
 *  @return 当前标签ID
 */
+ (NSUInteger)currentTag;

/**
 *  获取设备ID
 *
 *  @return 设备ID
 */
+ (nonnull NSString *)deviceId;

/**
 *  上报自定义异常
 *
 *  @param exception 异常信息
 */
+ (void)reportException:(nonnull NSException *)exception;

/**
 *  SDK 版本信息
 *
 *  @return
 */
+ (nonnull NSString *)sdkVersion;
```