iOS SDK集成手册

iOS SDK集成手册

本文档为阿里云iOS热修复方案集成手册。

【注意】由于存在苹果审核风险，iOS热修复方案可用于应用Bug修复，但不建议做日常业务升级迭代。

1 下载

- Demo

2 集成准备

SDK集成包括两部分:手动集成部分和Pods依赖部分，两者需要结合共同使用。

2.1 手动集成SDK

- 【注意】SDK文件为每个应用单独提供，用户每新建一个应用需首先获取应用的AppKey，然后请联系工作人员提供SDK文件。
- SDK包括:
  - AlicloudHotFixEMAS.framework
  - arp.framework

> 当前SDK版本:v1.0.5。(在AlicloudHotFixEmas.framework/Headers/AlicloudHotFixEmas.h中查看)

- 直接把下载SDK目录中的Framework拖入对应Target下即可，在弹出框勾选 `Copy items if needed` 。

2.2 Pod集成

- 指定Master仓库和阿里云仓库：

```
source 'https://github.com/CocoaPods/Specs.git'
source 'https://github.com/aliyun/aliyun-specs.git'
```

- 添加依赖：

```
pod 'AlicloudLua'
pod 'AlicloudUtils', '1.2.1'
pod 'ZipArchive', '~> 1.4.0'
```

（>为模糊指定版本号方式，> 1.0.0表明引用版本位于 `1.0.0 <= version < 1.1.0` 之间的最新版本SDK，用户可参考Podfile Syntax Reference，根据项目需要指定SDK版本。）

> 注意，若SDK集成过程中出现UTDID冲突，请参考：阿里云-移动云产品SDK UTDID冲突解决方案。

3 SDK接口使用说明

3.1 接入范例

获取服务实例调用应该尽可能的早，尽量在 `-[AppDelegate application:didFinishLaunchingWithOptions:]` 或者 `viewDidLoad` 的最开始进行SDK初始化操作，初始化之前不能用到其他自定义类，否则极有可能导致崩溃。而查询服务器是否有可用补丁的操作可以在后面的任意地方。

HotFix iOS SDK以全局 service 实例的方式提供服务，您可以通过以下方式获取实例：

专有云用户接口如下：

```
/**
 获取热修复实例
 */
```

```
+ (instancetype)sharedInstance;

/**
 初始化SDK

 @param appId            应用AppId
 @param callback         回调
 */
- (void)initWithAppId:(NSString *)appId
            appSecret:(NSString *)appSecret
             callback:(HotFixCallbackHandler)callback;

```

- 需从控制台获取 appId 及 appSecret

3.2 接口说明

3.2.1 初始化相关方法

设置补丁拉取时的App版本号：

```
/**
手动设置补丁拉取时的App版本号，SDK初始化前调用
(默认获取"CFBundleShortVersionString")

@param appVersion App版本号
*/
- (void)setAppVersion:(NSString *)appVersion;

/**
打开日志开关

@param enabled YES: 打开日志； NO：关闭日志
*/
- (void)setLogEnabled:(BOOL)enabled;

```

3.2.2 从服务端加载补丁方法

相关的接口如下：

```
/**
 从服务端加载补丁

 @param callback 回调
 */
- (void)loadPatch:(HotFixCallbackHandler)callback;

```

3.2.3 清空本地补丁方法

```
/**
清空本地补丁
*/
- (void)cleanPatch;

```

4 补丁生成

- 首先需要一个git仓库，在该仓库的根目录下创建一个名为 `patch` 的目录，同时仓库下需要放入我们提供的打包脚本 `build.sh`
- 参照[补丁脚本语法说明]，编写补丁文件，补丁文件名后缀一定是以 `.lua` 。
- 将一个或多个 `lua` 文件，存放到名为 `patch` 的文件夹中，【注意】必须放在 `patch` 目录下否则将导致patch文件找不到。目录格式如下：

```
- patch
  - patch1.lua
  - patch2.lua
  - ...
```

> 【注意事项】同一个版本中的最新Patch包总是要包括该版本所有的bug修复,举个例子：

- 新上线的版本1.0发现一个线上bug，标注为bug1；
- 制作patch.zip(补丁版本为1)，修复bug1；
- 之后该版本又发现一个bug，标注为bug2；
- 制作patch.zip(补丁版本为2)同时修复bug1和bug2，通过控制台上传补丁文件。

5 补丁发布

- 在Emas控制台操作发布补丁

6 调试工具使用

6.1 工具接入

- 指定Master仓库和阿里云仓库：

```
source 'https://github.com/CocoaPods/Specs.git'
source 'https://github.com/aliyun/aliyun-specs.git'
```

- 添加依赖

  专有云添加如下依赖：

```
pod 'AlicloudHotFixDebugEmas', '~> 1.0.5'
```

6.2 功能

- 调试工具提供 `加载本地补丁文件` 和 `扫码调试` 功能，方便在Patch上线前进行调试。

6.2.1 加载本地补丁文件

- 加载存储在本地的后缀名为 `.lua` 的补丁文件，如有多个文件，可多次调用。

```
#import <AlicloudHotFixDebug/AlicloudHotFixDebug.h>

[AlicloudHotFixDebugService loadLocalPachFile:patchFilePath];
```

6.2.2 扫码调试

在需要唤起扫码功能的 ViewController 内调用:

```
#import <AlicloudHotFixDebug/AlicloudHotFixDebug.h>

 // currentViewController 为需要唤起扫码功能的VC
[AlicloudHotFixDebugService showDebug:currentViewController];
```

- iOS 10 对隐私权限更加严格，需要添加权限说明(privacy description)。如果不做设置，可能会导致崩溃、审核不通过等情况。在info plist中增加下述配置，若二维码扫描页面崩溃，请您检查一下，是否添加了权限说明。

```
<key>NSCameraUsageDescription</key>
<string>访问相机</string>
```

- 扫码页面在控制台 - 补丁详情页面，补丁正式发布前，请在补丁灰度状态时，扫描二维码下载补丁进行本地测试。

【注意事项】

- 扫码加载Patch功能只能在真机下进行调试。

7 专有云服务端域名配置

```
/**
 设置服务地址
 */
- (void)setServerURL:(NSString *)url;
```