

监测SDK集成指南

v1.0

集成包介绍

libAHMonitor4AppStore.a - 静态库文件

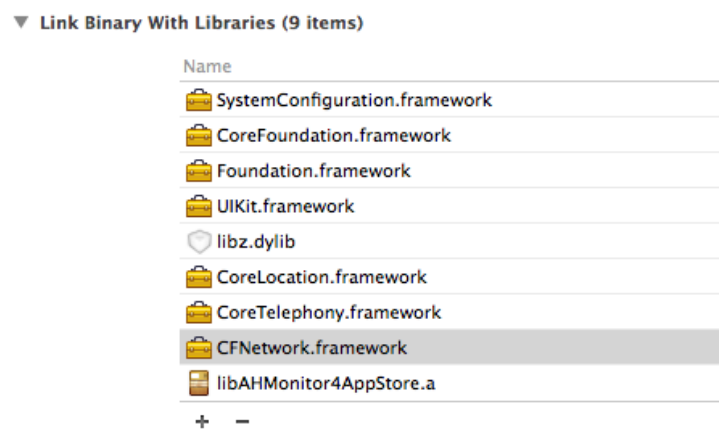
AHMonitor.h - SDK接口头文件

集成步骤

请把 libAHMonitor4AppStore.a 添加到 xcode 工程里“Build Phases”下的“Lnk Binary With Libraries”中。如果需要在模拟器环境下调试,请把 libAHMonitor4AppStore.a-i386.a 库也 导入到“Lnk Binary With Libraries”中。

必要的框架

本 SDK 需要依赖的框架如下图所示,请在 xocde 工程中添加未包含的框架。



接口描述

<pre>+(AHMonitor *)shareInstanceWithAppkey:(NSString *)appkey;</pre>
<ul style="list-style-type: none">• 描述:实例化方法• 参数:用来指定渠道唯一性的标示,此值需要由SDK提供方来指定• 返回:监测类实例
<pre>-(void)startMonitor;</pre>
<ul style="list-style-type: none">• 描述:启动监测运行• 参数:无• 返回:无
<pre>-(void)isEnableLog:(BOOL)yesOrNo;</pre>
<ul style="list-style-type: none">• 描述:是否打印出调试日志• 参数:• 返回:无
<pre>-(void)testMode:(BOOL)yesOrNo;</pre>

- 描述:启动测试模式(测试模式下,数据都将发往测试服务器)
- 参数:
- 返回:无

调用方法

1.在你工程中得Delegate里引入头文件

```
#import "AHMonitor.h"
```

2.在delegate的头文件添加QMMonitor全局变量

```
@property (strong, nonatomic) AHMonitor *monitor;
```

3.在delegate.m文件中的 - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions 添加启动代码:

```
//添加监测
self.monitor = [AHMonitor sharedInstanceWithAppkey:@"myapp"];
//在测试模式下运行
//[self.monitor testMode:YES];
//开启日志打印
//[self.monitor isEnabledLog:YES];
//启动监测
[self.monitor startMonitor];
```

4.如果你的项目中没有使用因为Background Fetch技术,则需要进一步的进行如下操作

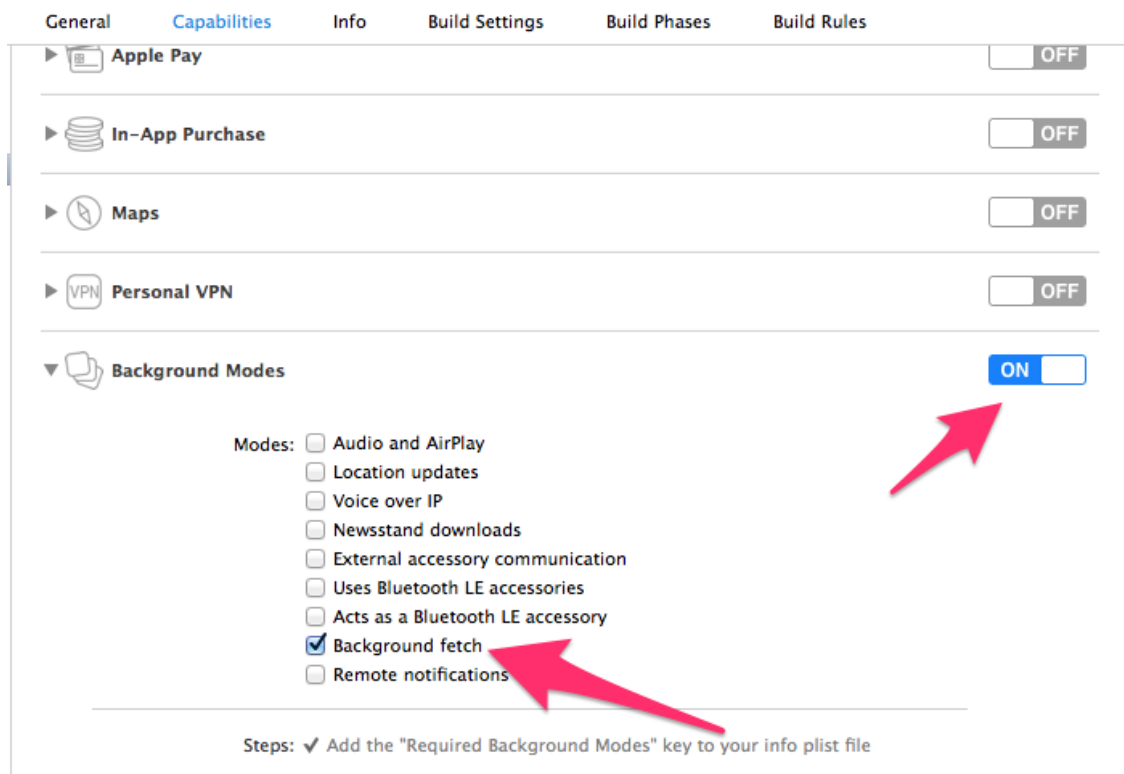
- 在(BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions方法中添加如下内容

```
[[UIApplication sharedApplication]
setMinimumBackgroundFetchInterval:UIApplicationBackgroundFetchIntervalMinimum];
```

- 同样我们需要在每次Background fetch执行后进行一次monitor工作,所以在delegate.m中添加如下方法:

```
-(void)application:(UIApplication *)application performFetchWithCompletionHandler:(void (^)(
UIBackgroundFetchResult))completionHandler
{
    [[QMMonitor sharedInstanceWithAppkey:@"myapp"] startMonitor];
    completionHandler(UIBackgroundFetchResultNewData);
}
```

- 最后,需要在工程中开启对Background fetch的支持,如下图



补充说明

- 调用本SDK不必采用异步执行启动,因为lib内部已使用异步执行方式

(完)