**iOS SDK Objective-c 安装⽂档**

# 1、登陆 OneAPM 后台

登录 SaaS Mi 后台 或 企业版 Mi后台（企业版后台地址请咨询 OneAPM 工作人员）

# **2、创建应⽤项目**

2.1.在应用程序列表,点击“创建“按钮。

2.2.输入应用名称,点击保存。

*注意：*

*包名不同的应用不可使用同一 Token。*

*应用名称可自定义，中文、英文、数字均可，建议不使用重复的 APP 名称，可在设置模块对已有的 APP 名称进行修改。*

2.3随后会获取应用的唯一标识 Token 值,用于配置 sdk 。

# **3、集成 SDK**

集成 Mi SDK 有两种方式：

* CocoaPods 配置 .framework
* 手动配置 .framework

## 3.1 CocoaPods 配置 .framework

3.1.1在 Podfile 文件添加：

pod 'OneAPMUEM','~> 5.5.2.4'

3.1.2在 Podfile 所在的文件夹下输入命令：

Pod install

3.1.3安装好后就可以打开 xcworkspace 文件，在你的项目中使用 iOS SDK 了。

open App.xcworkspace

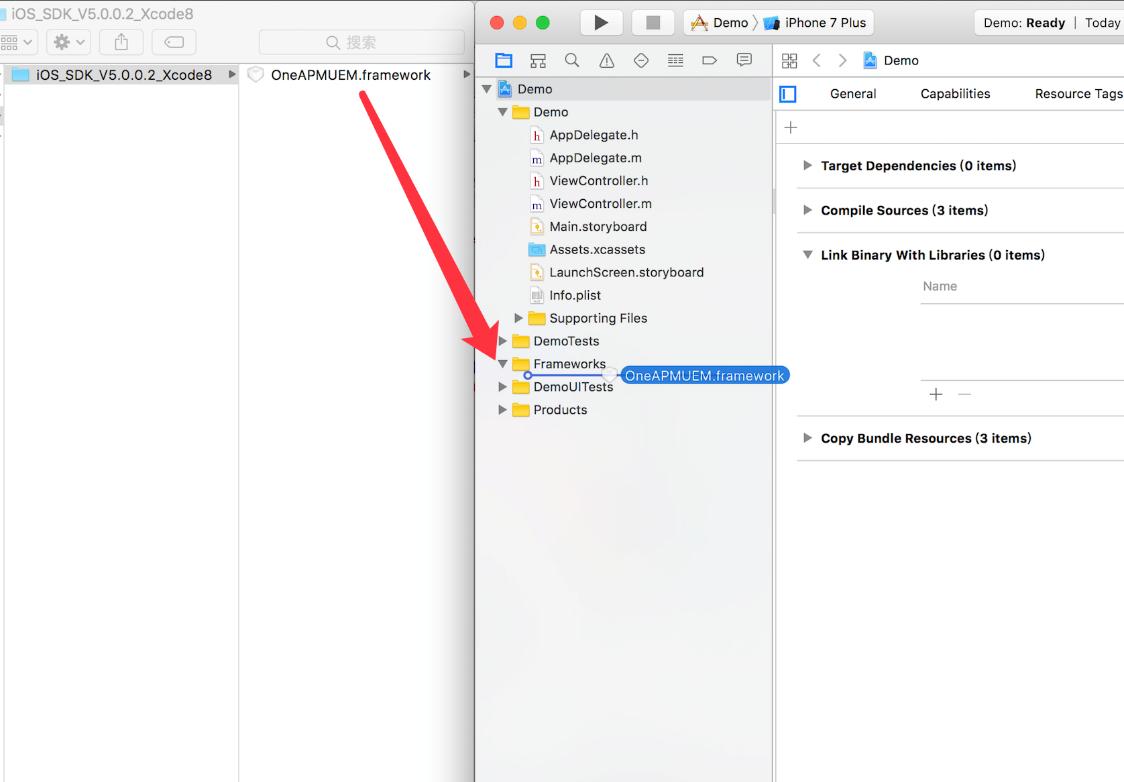
## 3.2 手动配置 .framework

### 3.2.1. 准备 iOS SDK。

从官网或 OneAPM 工作人员取得 SDK。

### **3.2.2. 添加 OneAPM Framework 至 Xcode 项目中**

解压 SDK,并将「OneAPMUEM.framework」文件夹从 Finder 中拖拽至 Xcode 项目中(悬停至导航窗口的项目中)。出现提示窗口时,选择「Copy items intodestination...」和「Create folder references...」。



### **3.2.3. 在 Linker Settings 中添加以下 5 个 Libraries**

在项目导航窗口内点击你的 Project，并选中你的 App，然后选择「Build Phases」选项卡。

打开「Link Binary with Libraries」列表。

点击添加:

* SystemConfiguration.framework
* CoreTelephony.framework
* libz.tbd\ libz.dylib (Xcode6 及以前版本)
* libstdc++.tbd\ libstdc++.dylib (Xcode6 及以前版本)
* WebKit.framework

# **4. 引入 SDK**

（1）在项目文件 [app\_name]-Prefix.pch（通常在文件夹「Support Files」中）或者 main.m 文件中，引入 OneAPM 头文件：

#import <OneAPMUEM/OneAPM.h>

1. 在 main.m 文件的 main 函数中添加如下代码:

int main(int argc, char \* argv[]) {

@autoreleasepool {

1.OneAPM log 开关设置（可选）

*// 可以通过这句话打印 OneAPM 的 Log，当及成以后出现类似 “OneAPM SDK init succeed.”的 Log，说明集成成功。*

[OneAPM setprintLog:YES];

2. 自定义配置用户信息（可选）

*// 为了更好的展示用户访问信息，开发者可以在 SDK 启动代码前通过 setUserName 来自定义设置用户的信息*

[OneAPM setUserName:@“phonenum”];

3. token 嵌入

*// 3.1 数据上传地址，若企业级服务器，没有提供对 HTTPS的支持，这里加上 withSecurity:NO*，如果是*HTTPS*则为 *withSecurity:YES 。*如有疑问详

*情可以咨询 OneAPM 工作人员：*

[OneAPM setHost:@"dc ip:port"withSecurity:NO];

*// 例如:123.123.123.123:8080*

*// 3.2 token 嵌入方式*

[OneAPM startWithApplicationToken:@" <use app token created at step 1>"];

Return UIApplicationMain (argc,argv,nil,NSStringFromClass

([AppDelegate class]));

}

}

*注意：*

*1. setUserName 信息用户可自定义，常见配置邮箱、电话等*

*2. 如未设置用户信息，用户信息列表默认展示 DeviceID*

*5. 运行应用程序并使用*

*Clean Project，并重新在模拟器或设备中启动应用程序，开始应用性能管理。*

*(注：开发时若要使用“崩溃统计”功能，请去掉 Debug executable 选项并在真机上运行，或者断开真机与 Mac 的连接)*

# **5.开启性能监控**

请静候 1 分钟，等待应用程序向 OneAPM 发送应用程序性能数据,即可开始使用OneAPM 应用性能管理功能。

# **6. 符号表（dSYM）上传**

为了正常解析崩溃信息，请按照以下步骤上传 dSYM 文件：

（1）在应用列表选择刚刚创建的应用，点击进入应用详情界面；

（2）选择“应用设置”模块，点击编辑符号化文件列表，即可上传对应版本的符号化文件；

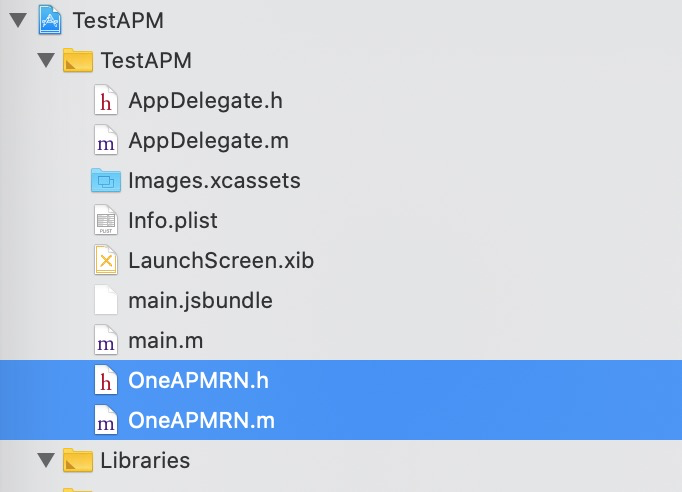
# **其他.兼容[React Native](http://www.baidu.com/link?url=Ioy5Yug_nfNAaAGHouly78SL6kNOyaNaGEwmuS0CuxXAFv05FdFk7ZxTBh_lgsDP" \t "/Users/qiaolizheng/Documents\\x/_blank) 框架的相关配置**

## 1. 集成 SDK。

参考前文中的步骤 3 集成 Mi SDK。

## 2. 导入文件

从下载的 SDK包中找到 ‘OneAPMRN.h’ ‘OneAPMRN.m’ 文件，拖入到工程中。



## 3. 初始化 SDK

在App.js文件中初始化OneAPM SDK，初始化代码如下：

import { NativeModules } from 'react-native';

*// 引⼊桥接⽂件*

OneAPMRN = NativeModules.OneAPMRN;

*// 为了更好的展示用户访问信息，开发者可以在 SDK 启动代码前通过 setUserName 来自定义设置用户的信息*

OneAPMRN.setUserName('OneAPMRN Test');

*// 数据上传地址，SaaS 用户无需配置，本地化部署用户需要根据自身服务器地址进行配置。另外若本地服务器不支持 HTTPS，需要将 true 改为 NO。*

OneAPMRN.setHost('mi dc ip',端口,true);

*// 根据 APP 实际情况调用，手动设置应用启动完成*

OneAPMRN.setAppStartFinish();

*// 初始化 SDK，嵌入 token*

OneAPMRN.startWithApplicationToken('your App Token');

*// 为确保 RN 应用页面被正常采集，通过以下接口方法对 RN 页面加载进行配置*

import { NativeModules } from 'react-native';

OneAPMRN = NativeModules.OneAPMRN;

*// ⻚⾯加载时，选择合适位置添加代码，注意更换 PageName*

OneAPMRN.onRNPageStart('PageName')

*// 举例：如放到Button的点击⾥，按钮点击时会上报⻚⾯加载数据*

<Button onPress = {() => OneAPMRN.onRNPageStart('PageName') } title = "⻚⾯开始" color = "#841584" />