

# Dreamobi SDK 集成文档

Dreamobi Technologies co.,Ltd.

Author: Nicky

Email: [nicky.wei@dreamobi.com](mailto:nicky.wei@dreamobi.com)

# 目录

- Dreamobi SDK 集成文档 ..... 1
- 1. Dreamobi SDK 介绍 ..... 3
- 2. 准备工作 ..... 3
  - 2.1 需要开发者提供 ..... 3
  - 2.2 下载 DreamobiSDK iOS 压缩包 ..... 3
- 3. 集成流程 ..... 3
  - 3.1 往工程中导入 DreamobiSDK 文件 ..... 3
  - 3.2 在 AppDelegate.m 文件中进行 DreamogiSDK 初始化操作 ..... 4
  - 3.3 调用视频墙广告（此步骤在 3.2 初始化成功以后） ..... 5
  - 3.4 调用直接播放视频广告（此步骤在 3.2 初始化成功以后） ..... 5
  - 3.5 NativeAd 的功能实现（此步骤在 3.2 初始化成功以后） ..... 6

# 1.Dreamobi SDK 介绍

Dreamobi SDK 为开发者提供高清、instant-play 的视频开发工具包，开发者可以利用视频墙、直接点击视频和 Native 三种展示形式带来远远高于其它广告形式的 ecpm。

## 2.准备工作

### 2.1 需要开发者提供

1. 应用在 APP Store 的下载链接
2. 应用的 bundle\_id
3. 集成哪种视频广告形式，a.点击直接播放 b.视频墙 c.NativeAd

### 2.2 下载 DreamobiSDK iOS 压缩包

1. 下载 DreamobiSDK-iOS 的链接 <https://github.com/Dreamobi/Dreamobi-iOS-SDK>
2. 你下载的压缩包解压后包含了下面几个文件



Document



Library



README.md

Document 文件夹中的 DreamobiSDK 的集成教程文档

Library 文件夹中是要添加到你的工程的文件，里面是 DreamobiSDK.framework.zip 文件，解压后是 DreamobiSDK.framework 静态文件添加到你的工程中。

## 3.集成流程

### 3.1 往工程中导入 DreamobiSDK 文件

1. 往工程中导入 DreamobiSDK.framework 静态文件
2. 为工程添加系统自带的 Frameworks
  - MediaPlayer.framework
  - SystemConfiguration.framework

- AdSupport.framework

### 3.2 在 AppDelegate.m 文件中进行 DreamogiSDK 初始化操作

1. 添加 Dreamobi SDK 头文件，在 AppDelegate.m 文件中添加  
`#import <DreamobiSDK/DreamobiConfigure.h>`
2. 在 AppDelegate.m 自动生成的

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
```

的函数里面添加下面代码

```
[[DreamobiConfigure config]configureWithAppToken:@"Your AppToken" zoneIDs:@[@"ZoneId"] appBundleId:@"Your AppBundleId" delegate:self];
```

注意替换：Your AppToken（NSString 类型）、ZoneId（NSArray 类型）和 Your AppSecret（NSString 类型）为你的 appToken、zoneIds 和 appBundleId

参数说明：

App_token	用于进行权限验证
Bundle_id	应用在 APP store 的 bundle_id
zoneIds	广告位（以数组组的形式 必须要）

当 delegate 中的参数为 self 时，遵守 DreamobiAdVideoRewardDelegate 协议，并要实现其代理方法，在 AppDelegate.m 文件中添加下面代码

```
#import "AppDelegate.h"
#import <DreamobiSDK/DreamobiConfigure.h>

@interface AppDelegate ()<DreamobiAdVideoRewardDelegate>
@end
```

代理方法的函数

```
-(void)dreamobiAdVideoReward :(BOOL)success isConnect:(BOOL)isConnect getPoint:(long)point
{
    NSLog(@"回调成功");

    if (success) {
        NSLog(@"视频成功播放返回用户金币%d",point);
    }
    if (isConnect) {
        NSLog(@"网络能支持播放视频");
    }
}
```

该函数是实现展示 wallAd 和 videoAd 回调方法（开发者自定义逻辑）

如果不需要回调逻辑的话 delegate 的参数为 nil，以上逻辑忽略。

功能简介：Dreamobi SDK 会在用户播放视频后，返回不同的状态，开发者可以根据状态实现对应的处理逻辑，如给用户增加金币等

参数说明：

Success	标识视频是否播放成功
---------	------------

isConnect	当前用户是否联网
point	返给用户的金币数量

### 3.3 调用视频墙广告（此步骤在 3.2 初始化成功以后）

1. 在开发者所要展示的界面文件中加入 Dreamobi SDK 头文件  
#import <DreamobiSDK/DreamobiConfigure.h>头文件
2. 在需要展示视频墙的地方调用下面的函数

```
[DreamobiConfigure showAdWallWithZone:@" " withDelegate:self];
```

注意：当开发者不需要视频开始和结束的状态时，开发者可以在上述方法 withDelegate 的参数中传 nil，并忽略步骤 3 如下 其中 zoneId 为 Configure 中对应广告形式 zoneID

```
[DreamobiConfigure showAdWallWithZone:@" " withDelegate:nil];
```

3. 遵守 DreamobiAdDelegate 协议（当 2 中方法传递不为 nil 需要实现此步骤）  
需要在实现视频墙的文件中实现 DreamobiAdDelegate 协议

```
#import "ViewController.h"
#import <DreamobiSDK/DreamobiConfigure.h>

@interface ViewController ()<DreamobiAdDelegate>
```

- A. 实现开始播放视频的回调

```
- (void)dreamobiAdVideoStarted
{
    NSLog(@"开始播放视频 回调成功");
}
```

- B. 实现视频播放结束的回调

```
- (void)dreamobiAdVideoFinished
{
    NSLog(@"播放视频结束 回调成功");
}
```

### 3.4 调用直接播放视频广告（此步骤在 3.2 初始化成功以后）

1. 在调用视频所在界面的文件中添加下头文件  
#import <DreamobiSDK/DreamobiConfigure.h>头文件
2. 判断本地是否有可播放的视频 方法如下 返回 BOOL 值 YES 代表有 NO 代表没有视频

```
[DreamobiConfigure isAvailVideoWithZoneID:@""];
```

3. 在调用视频的地方，实现下面函数

```
[DreamobiConfigure showVideoWithZone:@"" withDelegate:self];
```

注意：当开发者不需要视频开始和结束的状态时，开发者可以在上述方法中传 nil, 参照 [3.3-2](#) zoneID 为 Configure 中对应广告形式 zoneID

4. 遵守 DreamobiAdDelegate 协议,参照 [3.3-3](#)
5. 此功能为视频先缓存在本地，如果需要展示缓存视频的相关信息执行如下方法

```
DreamobiVideo *videoMessage = [[DreamobiVideo alloc] init];  
videoMessage = [DreamobiConfigure getVideoInfoWithZoneID:@""];
```

zoneID 为需要直接播放视频对应 zoneID

执行此方法的得到的是 DreamobiVideo 对象 其属性里有相关的视频信息

### 3.5 NativeAd 的功能实现（此步骤在 3.2 初始化成功以后）

1. 请求 NativeAdMessage 的实现函数

```
[DreamobiConfigure getNativeAdWithZone:@"I" withIconSize:NOSIZE withBannerSize:NOSIZE withDelegate:self];
```

zoneID 为 configure 时对应广告形式 zoneID

withDelegate 传 self 执行回调获取 NativeAd 需要遵守 2 中协议

如果为 nil 不执行回调

IconSize 和 BannerSize 没尺寸要求时 传 NOSIZE

如有尺寸要求 尺寸有以下几种

```
...  
|  
NOSIZE=0,  
  
ICON_50X50 =1,  
ICON_100X100 =2,  
ICON_175X175 =3,  
BANNER_1200X627 =4,  
BANNER_640X334 =5,  
...  
- - - -
```

2. 请求 NativeAd 回调需要遵守 DreamobiNativeAd 协议（用来回调 nativead 信息）

```
@interface ViewController ()<DreamobiNativeAd>
```

回调要实现的方法

```
- (void)dreamobiNativeAd:(DreamobiNative *)nativeMessage  
{  
  
    NSLog(@"native回调-----%@", nativeMessage);  
}
```

回调返回的是 DreamobiNative 的对象 其属性是 nativeAd 的相关信息  
如果没有请求到数据的时候 回调 DreamobiNative 对象为空