

猎豹移动天马广告 iOS-SDK 使用指南

(V1.5 版本)

目录

前提条件.....	2
1.为您的应用创建 MID 和 POSID	2
2.将 SDK 添加到 Xcode 项目中 (手动方式)	2
3.初始化 CheetahMobileAds	5
4.创建您的原生广告.....	6
5.创建您的横幅广告.....	8
6.创建您的插屏广告请求	11
7.创建您的开屏广告请求	14
8.错误码说明	17

前提条件

- Xcode5.1 或更高版本
- 部署目标为 7.0 或更高版本

1.为您的应用创建 MID 和 POSID

请到 <http://pub.adkmob.com/index.php> 为您的应用创建 MID 和 POSID

(请先向商务申请 ssp 平台账号，商务会为您开户。)

2.将 SDK 添加到 Xcode 项目中 (手动方式)

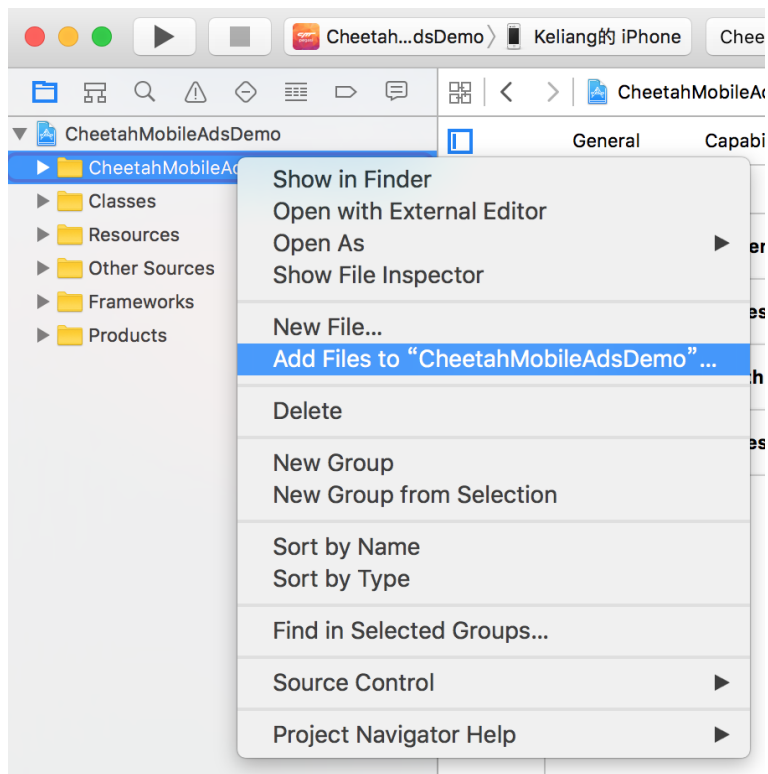
如果您还没有下载 CheetahMobileAds，请前往 Github

(<https://github.com/CMAAdSDK/pegasi-ios-sdk>) 进行下载并解压

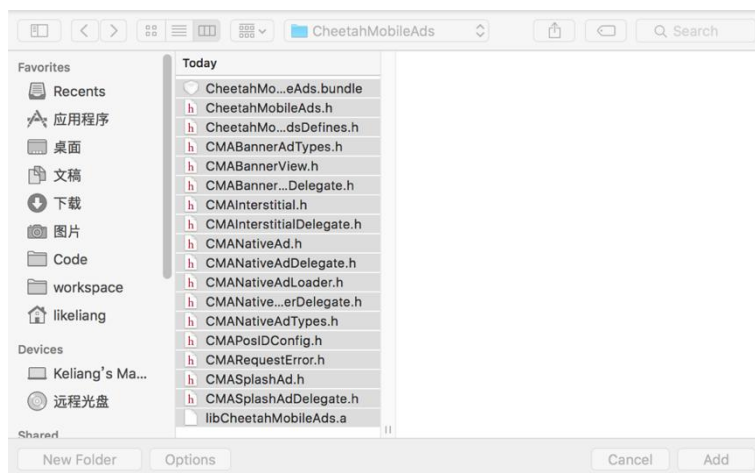
步骤 1：添加 SDK 文件到 Xcode 项目中

右键点击 CheetahMobileAdsDemo 项目，然后选择 “Add Files To

CheetahMobileAdsDemo” (将文件添加到 CheetahMobileAdsDemo)。



添加 CheetahMobileAds 文件夹下的 SDK 文件：



步骤 2：添加 SDK 需要的其它框架

SDK 需要以下 iOS 开发框架，您的项目中可能不包含这些框架：

- AdSupport
- CoreTelephony
- StoreKit

- CoreLocation
- Security
- SystemConfiguration

选择 Project Settings > Build Phases

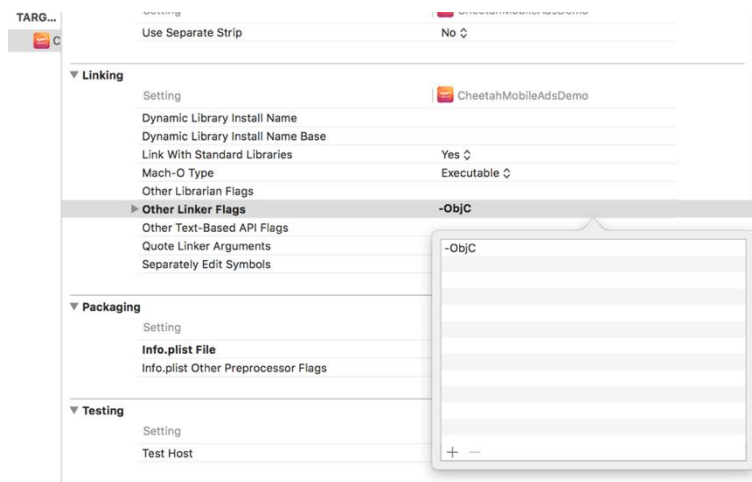
展开 "Link Binary with Libraries" 选项

点击 "+", 添加其它需要的框架

步骤 3 : 更改静态库设置

点击项目 Target>Build Settings>Linking

找到下面的 "Other Linker Flags" 添加-ObjC (如下图所示)



步骤 4 : 修改 info.plist 中 AppTransportSecurity 配置 (针对 iOS 9)

由于 iOS9 引入了 AppTransportSecurity(ATS)特性，所以需要开发者在工程中增加

设置 NSAppTransportSecurity 以支持可以发送 http 请求，配置如下图

Bundle version	String	1
Application requires iPhone enviro...	Boolean	YES
▼ App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES
Launch screen interface file base...	String	LaunchScreen
► Required device capabilities	Array	(1 item)
Status bar is initially hidden	Boolean	NO

3.初始化 CheetahMobileAds

在您的 AppDelegate.h 中引用 CheetahMobileAds

```
#import "CheetahMobileAds.h"
```

在程序启动时为 CheetahMobileAds 配置 MID

在-application: didFinishLaunchingWithOptions:方法中插入以下代码

```
[CheetahMobileAds startWithMIDConfig:[[CMAMIDConfig alloc]  
initWithMID:@"YOUR_MID" chinaMID:@"YOUR_MID_INCHINA"]];
```

4.创建您的原生广告

步骤 1 : 在 NativeAdDetailsViewController 中声明 CMANativeAdLoader 对象 , 以及 CMANativeAd 对象

```
@interface NativeAdDetailsViewController ()<CMANativeAdLoaderDelegate>

@property (nonatomic) CMANativeAdLoader *adLoader;

@property (nonatomic) CMANativeAd *nativeAd;

@end
```

步骤 2 : 初始化 CMANativeAdLoader

```
- (void)viewDidLoad {

    [super viewDidLoad];

    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]
initWithPosID:@"1345103" chinaPosID:@"1346103"];

    self.adLoader = [[CMANativeAdLoader alloc]
initWithPosIDConfig:config adTypes:@[ kCMANativeAdLoaderNewsFeed]];

    self.adLoader.delegate = self;

}
```

步骤 3 : 在需要加载广告的地方调用-loadAd 方法

```
[self.adLoader loadAd];
```

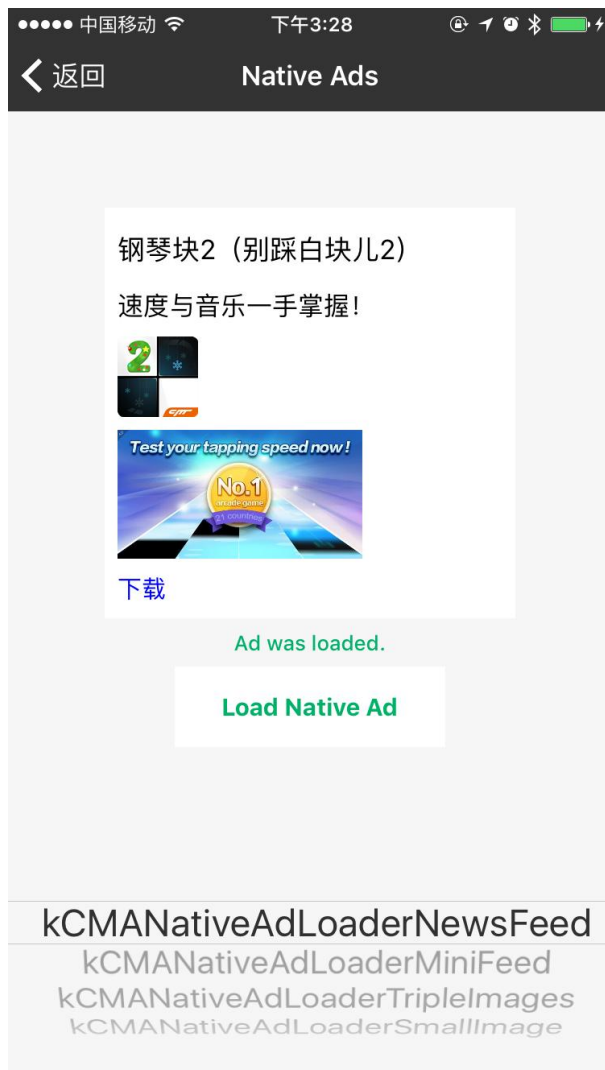
步骤 4 : 实现 CMANativeAdLoaderDelegate 方法 , 来跟踪广告生命周期中的各个

阶段（可选）

```
// 加载广告成功后回调  
- (void)nativeAdLoaderLoaded:(CMANativeAdLoader *)nativeAdLoader;  
  
// 加载广告失败后回调  
- (void)nativeAdLoader:(CMANativeAdLoader *)nativeAdLoader  
didFailToReceiveAdWithError:(CMARequestError *)error;
```

结果：

以下是 Demo 中展示原生广告的广告效果



5.创建您的横幅广告

您可以通过 Storyboard 或代码创建这个 BannerView , 这里是用代码创建一个 320 x 50 的 BannerView.

步骤 1. 在 BannerAdViewController 中声明 CMABannerView 对象

```
@interface BannerAdViewController ()<CMABannerViewDelegate>

@property (strong, nonatomic) CMABannerView *banner;

@end
```

步骤 2. 在-viewDidLoad:中初始化 CMABannerView

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]
initWithPosID:@"1345105" chinaPosID:@"1346105"];

    self.banner = [[CMABannerView alloc]
initWithBannerAdType:kCMABannerAdTypeBanner];

    self.banner.delegate = self;

    self.banner.posIDConfig = config;

    self.banner.rootViewController = self;
}
```


步骤 3. 调用- loadAd 方法把广告加载到 BannerView 中

```
- (IBAction)didClickedLoadAdBtn:(id)sender
{
    [self.banner loadAd];

    self.adStatusLabel.text = @"Requesting ad...";
}
```

步骤 4. 实现 CMANativeAdLoaderDelegate 方法，来跟踪广告生命周期中的各个阶段（可选）

```
// 广告加载成功时回调
- (void)bannerViewDidLoadAd:(CMABannerView *)banner;

// 广告加载失败时回调
- (void)bannerView:(CMABannerView *)banner
didFailToLoadAdWithError:(CMARquestError *)error;

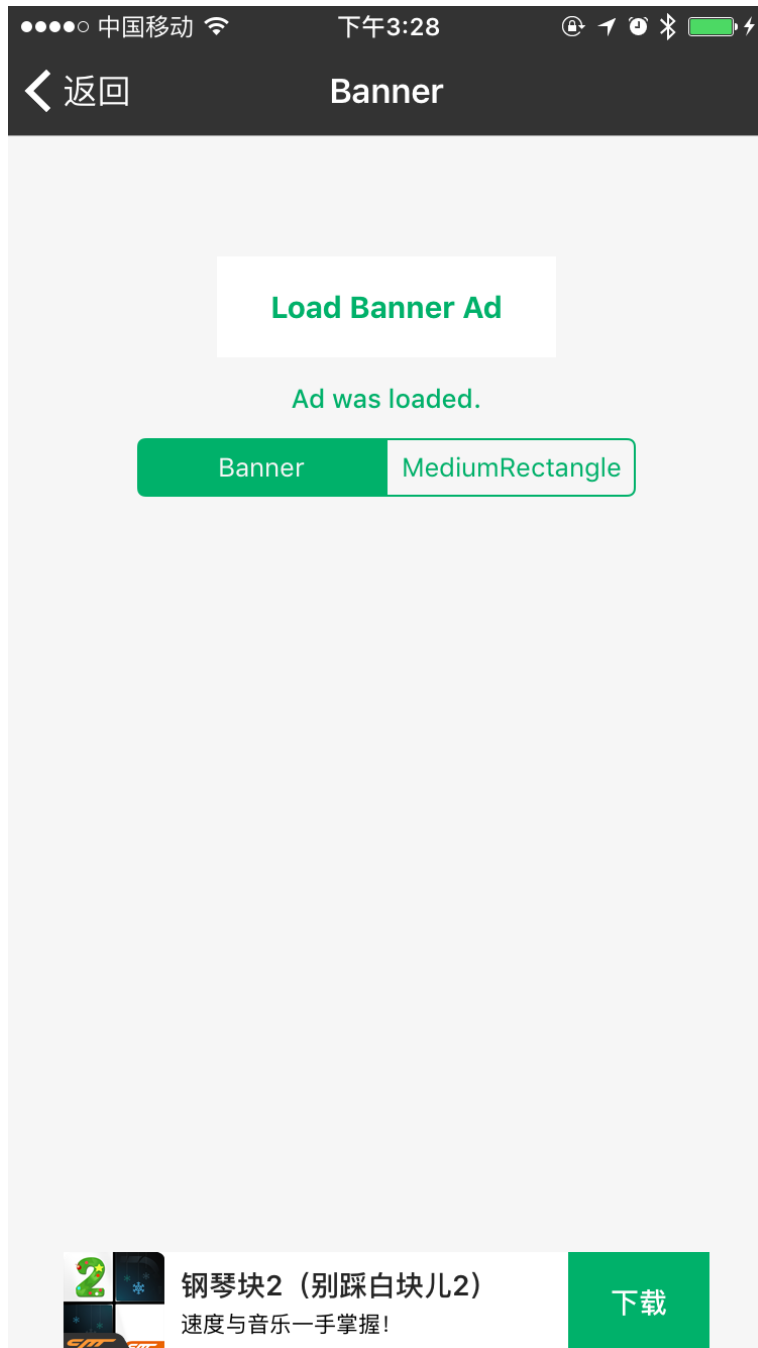
// 点击广告弹出网页视图前回调
- (void)bannerViewWillPresentScreen:(CMABannerView *)banner;

// 网页视图被关闭前回调
- (void)bannerViewWillDismissScreen:(CMABannerView *)banner;

// 点击广告离开应用前回调
- (void)bannerViewWillLeaveApplication:(CMABannerView *)banner;
```

结果：

以下是 Demo 中展示横幅广告效果



6.创建您的插屏广告请求

步骤 1. 在 AppDelegate 中声明 CMAInterstitial 变量

```
@interface InterstitialAdViewController ()<CMAInterstitialDelegate>

@property (nonatomic) CMAInterstitial *interstitial;

@end
```

步骤 2. 在-viewDidLoad:中初始化 CMAInterstitial 变量

```
- (void)viewDidLoad {

    [super viewDidLoad];

    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]
initWithPosID:@"1345104" chinaPosID:@"1346104"];

    self.interstitial = [[CMAInterstitial alloc] initWithPosIDConfig:config];

    self.interstitial.delegate = self;

}
```

步骤 3. 在需要加载广告的时机调用- loadAd 方法

```
- (IBAction)didClickedLoadAdBtn:(id)sender

{
```

```
self.adStatusLabel.text = @"Requesting ad...";

[self.interstitial loadAd];

}
```

步骤 4. 实现 CMAInterstitialDelegate 方法，来跟踪广告生命周期中的各个阶段 (可选)

```
//加载广告成功后回调
- (void)interstitialDidLoadAd:(CMAInterstitial *)ad;

//加载广告失败后回调
- (void)interstitial:(CMAInterstitial *)ad
didFailToLoadAdWithError:(CMARequestError *)error;

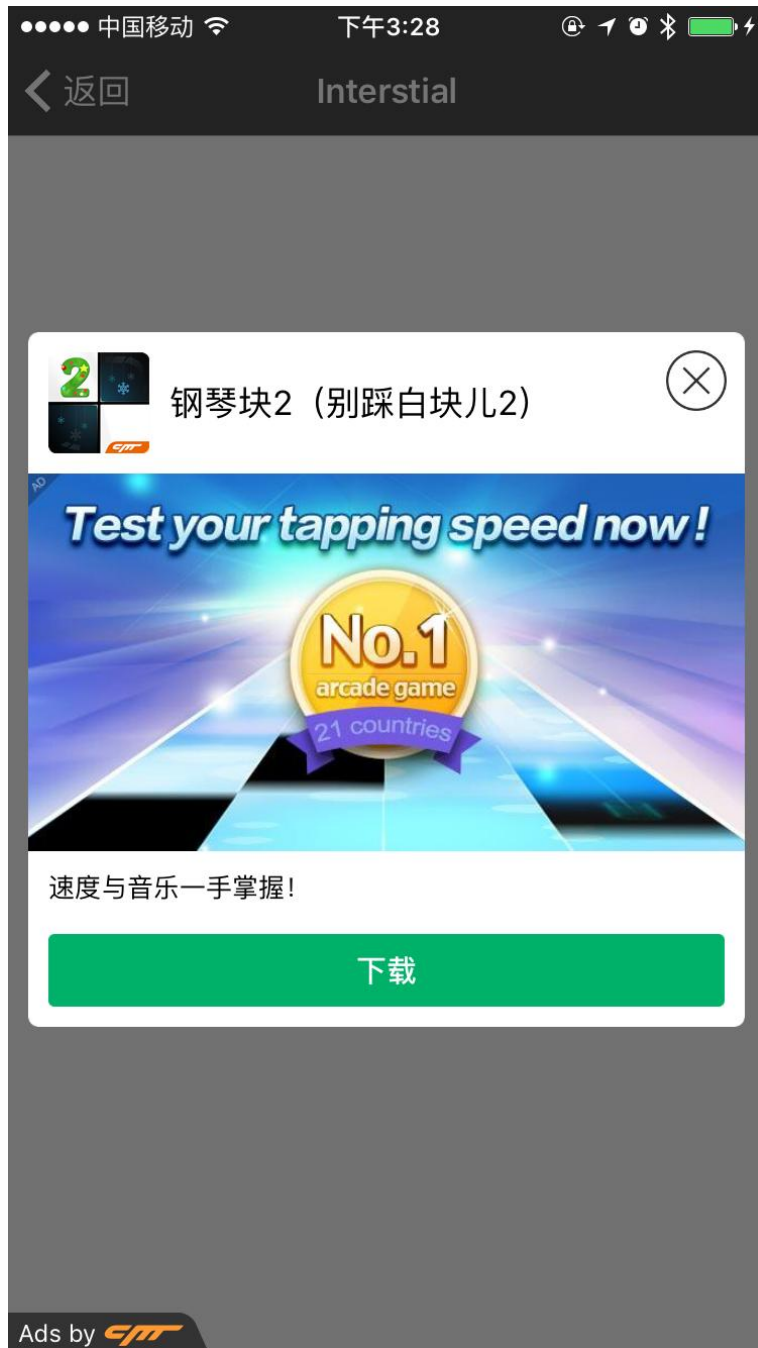
//插屏广告弹出前回调
- (void)interstitialWillPresentScreen:(CMAInterstitial *)ad;

//插屏广告推出前回调
- (void)interstitialWillDismissScreen:(CMAInterstitial *)ad;

//点击广告后离开应用前回调
- (void)interstitialWillLeaveApplication:(CMAInterstitial *)ad;
```

结果：

以下是 Demo 中插屏广告效果



7.创建您的开屏广告请求

步骤 1. 在-application: didFinishLaunchingWithOptions:方法中加入加载开屏

代码

```
didFinishLaunchingWithOptions method

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

    [CheetahMobileAds startWithMIDConfig:[[CMAMIDConfig alloc]
initWithMID:@"1345" chinaMID:@"1346"]];

    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]
initWithPosID:@"1345106" chinaPosID:@"1346106"];

    CMASplashAd *splashAd = [[CMASplashAd alloc]
initWithPosIDConfig:config];

    splashAd.delegate = self;

    splashAd.dismissAnimation = kCMASplashAnimationGrowFade;

    splashAd.backgroundView = [self splashBackgroundView];

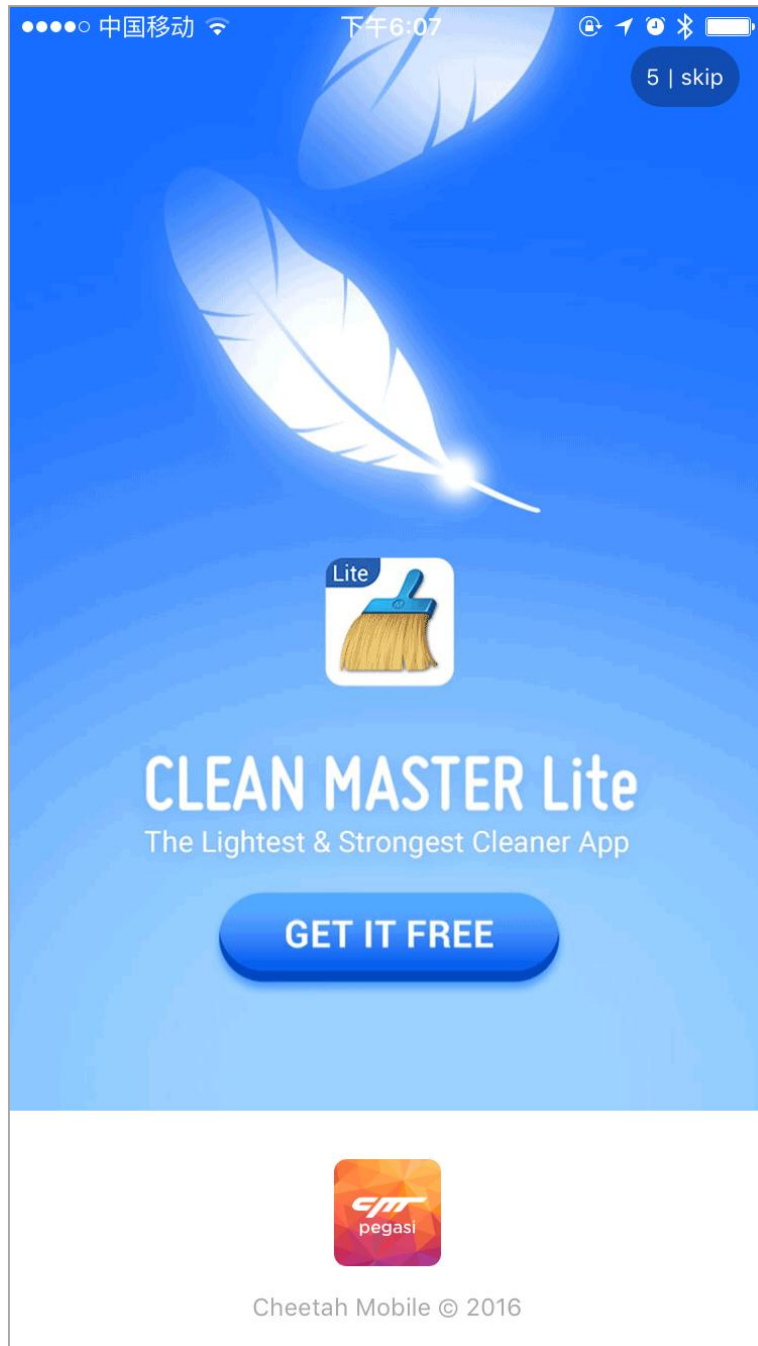
    [splashAd show];
}
```

步骤 2. 实现 CMASplashAdDelegate , 来跟踪广告生命周期中的各个阶段 (可选)

```
// When ad loading succeeds, callback:  
  
- (void)splashAdDidLoadAd:(CMASplashAd *)splashAd;  
  
//When ad loading fails, callback:  
  
- (void)splashAd:(CMASplashAd *)splashAd  
didFailToLoadAdWithError:(CMARquestError *)error;  
  
//Before the display of splash screen, callback  
  
- (void)splashAdWillPresentScreen:(CMASplashAd *)splashAd;  
  
//Before the disappear of splash screen, call back  
  
- (void)splashAdDidDismissScreen:(CMASplashAd *)splashAd;  
  
// ( When the ad is clicked, ) callback before leaving the application:  
  
- (void)splashAdWillLeaveApplication:(CMASplashAd *)splashAd;  
  
//When the “skip” button is clicked, callback:  
  
- (void)splashAdWillSkip:(CMASplashAd *)splashAd;
```

结果：

以下是 Demo 中开屏广告效果



8.错误码说明

-1001	内部错误
-1002	拉取广告超时错误
-1003	拉取广告时网络环境错误
-1004	没有广告填充
-1005	JSON 解析错误
-1006	服务器错误
-1007	请求参数错误
-1008	资源文件下载错误