
腾讯广告联盟 **iOS SDK**

接入说明文档

V4.7.3 版

法律声明

客户在注册成为腾讯广告联盟用户之后，联盟向客户提供本文档，作为其与联盟广告平台对接的指引文档。腾讯广告联盟拥有修改、调整、增补本文件的权利，并在法律允许范围内对本文档拥有最终解释权。

修订历史

| 文档版本 | 修订日期 | 修订说明 |
|--------|------------|---|
| V4.7.3 | 2017-4-17 | 修复 Bug |
| V4.7.2 | 2017-3-30 | 修复 Bug |
| V4.7.1 | 2017-3-7 | 优化开屏广告效果 |
| V4.7.0 | 2017-12-10 | 增加原生视频模板、适配 iPhone X、支持 Auto Layout |
| V4.6.4 | 2017-10-25 | 优化广告位内存占用 |
| V4.6.3 | 2017-10-11 | 增加原生广告三小图接口 |
| V4.6.2 | 2017-08-29 | 增加原生广告接口 |
| V4.6.1 | 2017-07-28 | 修改返回的错误码 |
| V4.6.0 | 2017-07-13 | 增加原生模板广告形态支持，增加 HTTPS 支持 |
| V4.5.6 | 2017-03-1 | 修复 Bug，增加错误码说明 |
| V4.5.5 | 2017-01-16 | 优化插屏视频落地页样式 |
| V4.5.4 | 2016-12-20 | 优化数据上报。 |
| V4.5.2 | 2016-12-2 | 优化数据上报，开屏回调接口，打开广告的体验优化。 |
| V4.5.1 | 2016-11-2 | 最低支持 iOS7 系统，优化数据上报 |
| V4.5.0 | 2016-10-08 | 开屏跳过按钮支持自定义，广告标识字样修改 |
| V4.4.9 | 2016-9-26 | 修复 Bug。 |
| V4.4.8 | 2016-9-12 | 优化数据上报，调整原生广告拉取上限 |
| V4.4.7 | 2016-8-25 | 增加广告标识 |
| V4.4.6 | 2016-8-18 | 优化广告体验 |
| V4.4.5 | 2016-8-1 | 优化开屏回调接口 |
| V4.4.3 | 2016-6-21 | 增加广告有效期的说明 |
| V4.4 | 2016-5-23 | 优化了广告点击体验 |
| V4.3.1 | 2016-4-18 | 移除 ObjC 链接依赖 |
| V4.3 | 2016-2-20 | SDK 支持半屏开屏功能。 |
| V4.2 | 2016-2-2 | SDK 支持转化统计功能，优化点击行为。 |
| V4.1 | 2015-11-3 | SDK 支持 Bitcode，去除 c++ 链接依赖，不再对 iOS4,5 作支持。 |
| V4.0 | 2015-8-31 | 按照统一模版撰写 V4.0 版文档。 支持 Banner、插屏、原生广告、开屏。 |

目录

| | |
|--------------------------------|-----------|
| 1. 概述 | 3 |
| 2. 背景 | 3 |
| 2.1. 开发环境 | 3 |
| 2.2. 术语介绍 | 3 |
| 3. 接入流程 | 3 |
| 3.1. 广告接入流程 | 4 |
| 3.1.1. 新建媒体 | 4 |
| 3.1.2. 新建广告位 | 4 |
| 3.2. 转化统计接入流程 | 5 |
| 4. 接入配置 | 5 |
| 4.1. 添加 SDK 到工程中 | 5 |
| 4.2. 为工程添加相应的 Frameworks | 7 |
| 5. 接入代码 | 8 |
| 5.1. Banner 广告接入代码 | 8 |
| 5.2. 插屏广告接入代码 | 10 |
| 5.3. 原生广告接入代码 | 12 |
| 5.4. 开屏广告接入代码 | 12 |
| 5.5. 原生模板功能接入代码 | 14 |
| 5.6. 转化统计功能接入代码 | 错误!未定义书签。 |
| 6. 问题排查 | 18 |
| 7. 注意事项 | 19 |

1. 概述

本文档在帮助 iOS 应用开发者在程序中快速植入腾讯广告联盟平台提供的广告的同时，提供给广告主帮助其统计应用的转化统计数据。作为应用开发者，您只需要进行简单配置，就可以在您的应用中显示定制的广告或统计转化数据。关于 SDK 的具体使用方法，请仔细阅读下面的文档。

2. 背景

2.1. 开发环境

操作系统：Mac OS X 10.13.0 及以上版本

开发工具：Xcode 9 及以上版本

部署目标：iOS 7.0 及以上版本

支持设备：iPhone/iPod Touch/iPad

2.2. 术语介绍

APPID: 媒体 ID，是您在广点通移动联盟官网创建媒体时获得的 ID，这个 ID 是我们在广告网络中识别您应用的唯一 ID。

POSID: 广告位 ID，是您在广点通移动联盟官网为您的应用所创建的某种类型（Banner、开屏、插屏、原生）的广告位置的 ID。

3. 接入流程

SDK 接入流程分成广告功能接入流程和转化统计功能接入流程两部分。对于想要嵌入广告功能的用户，需要开发者在联盟系统里注册会员。而对于想要嵌入转化功能的用户，可以跳过这一步骤。

3.1. 广告接入流程

在嵌入SDK之前首先需要您在广点通移动联盟官网<http://e.qq.com/dev>中注册成为联盟会员。注册完成会员后在联盟系统创建媒体和广告位相关信息并获得 APPID 和 POSID，创建流程如下：

3.1.1. 新建媒体

请在移动联盟媒体管理界面选择新建媒体，输入具体的媒体名称、类别等相关信息，即可完成创建媒体的流程。如下图：

新建媒体

媒体类型：☐ Android程序 ☒ iOS程序

媒体名称：

关键词：

媒体类别：移动游戏 赛车

功能简介：

下一步 取消

3.1.2. 新建广告位

在您新建完成媒体之后，可以在广告位管理页面选择“创建广告位”。创建广告位界面如下图所示：

新建广告位

×

媒介选择：

天天酷跑

▼

广告位名称：

广告位类型：

☒ Banner广告

☐ 应用墙

☐ 插屏广告

☐ 开屏广告

☐ 原生广告

创建

取消

步骤说明：

1. 选择要嵌入广告的媒体名称（未审核通过应用下的广告位无法拉取到广告）
2. 设置广告位的名称及格式。广告位名称用于区分应用内设置的各个广告位，方便您在后台进行统一管理和查看数据报表。
3. 点击创建按钮以后即获得对应该广告位的 ID 号。
4. 成功得到广告位 ID 且应用通过审核以后，您可以在应用代码内使用该 ID 进行广告联调测试（见“接入代码”部分）。

以上各环节的运营接口人联系方式：GDTLM@tencent.com

备注：

在新建插屏广告位时如果应用为横屏应用，不可勾选大插屏尺寸。大插屏尺寸只适用于竖屏应用。

3.2. 转化统计接入流程

想要接入转化统计功能的用户，可以跳过在广告联盟注册，创建广告位等步骤，直接参考 Sample 示例或是下述第 4 节，第 5.5 两章内容来完成接入。

4. 接入配置

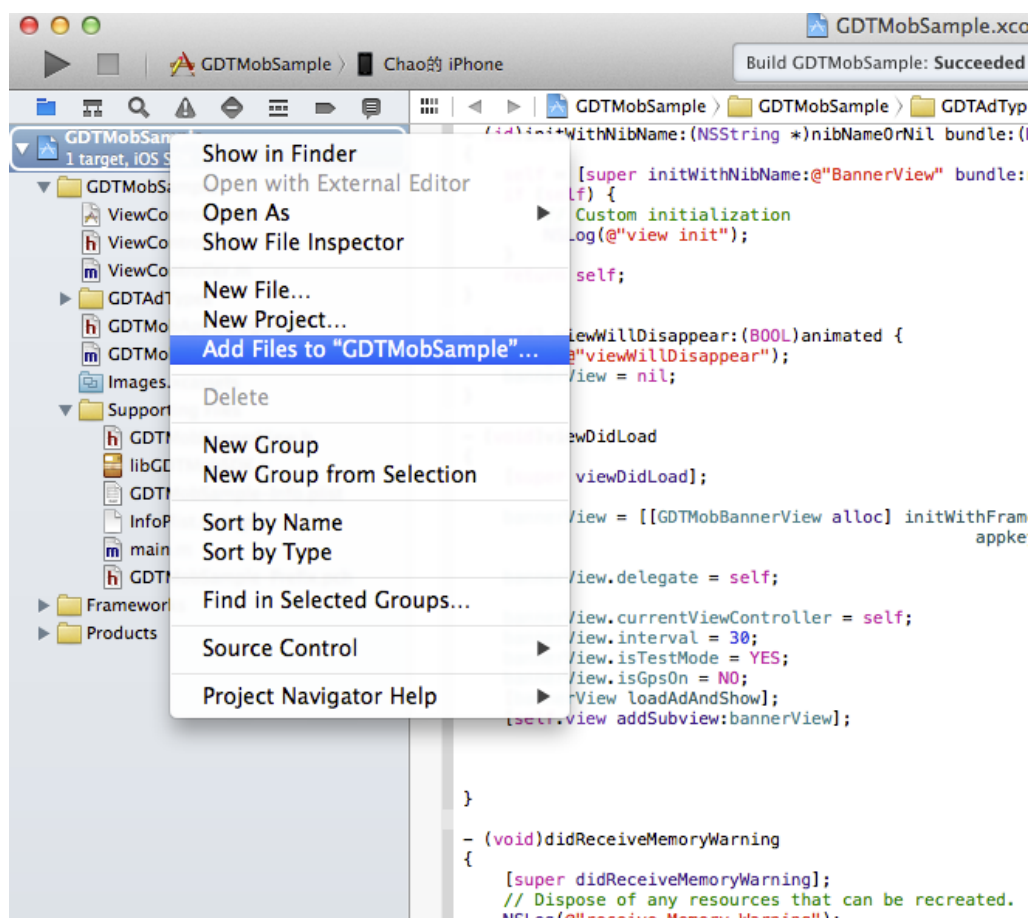
4.1. Build Settings

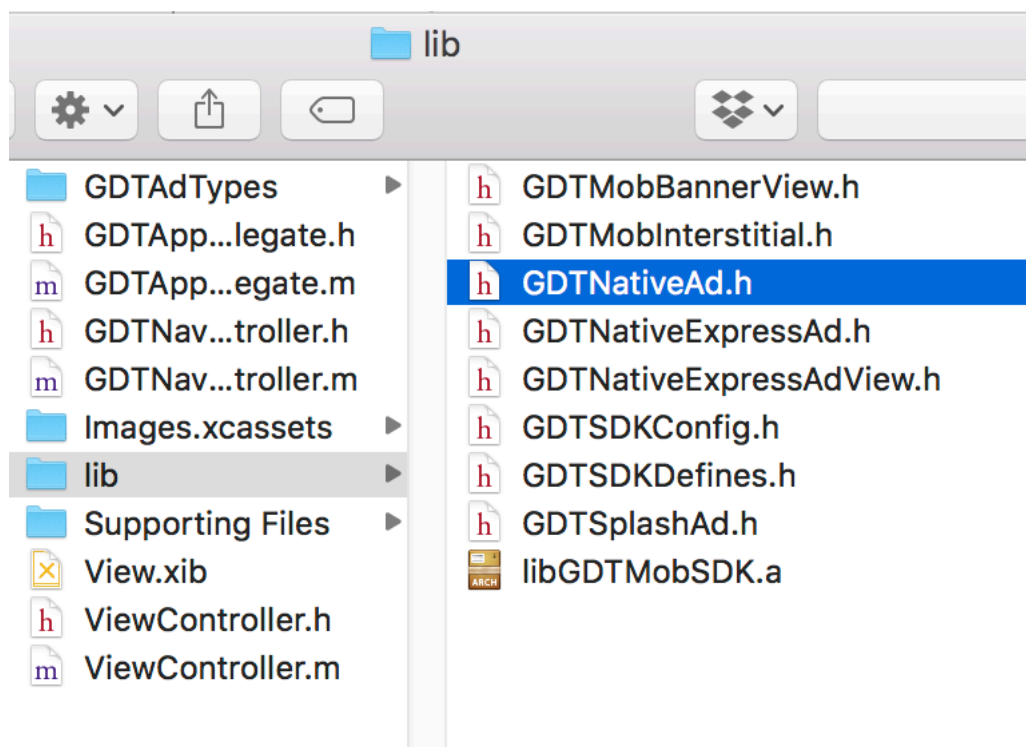
Other Linker Flags 添加 -ObjC

4.2. 添加 SDK 到工程中

请在自己的工程导入libs文件夹下的SDK文件：

- libGDTMobSDK.a
- GDTMobBannerView.h
- GDTNativeExpressAd.h
- GDTNativeExpressAdView.h
- GDTMobInterstitial.h
- GDTNativeAd.h
- GDTSplashAd.h
- GDTSDKDefines.h

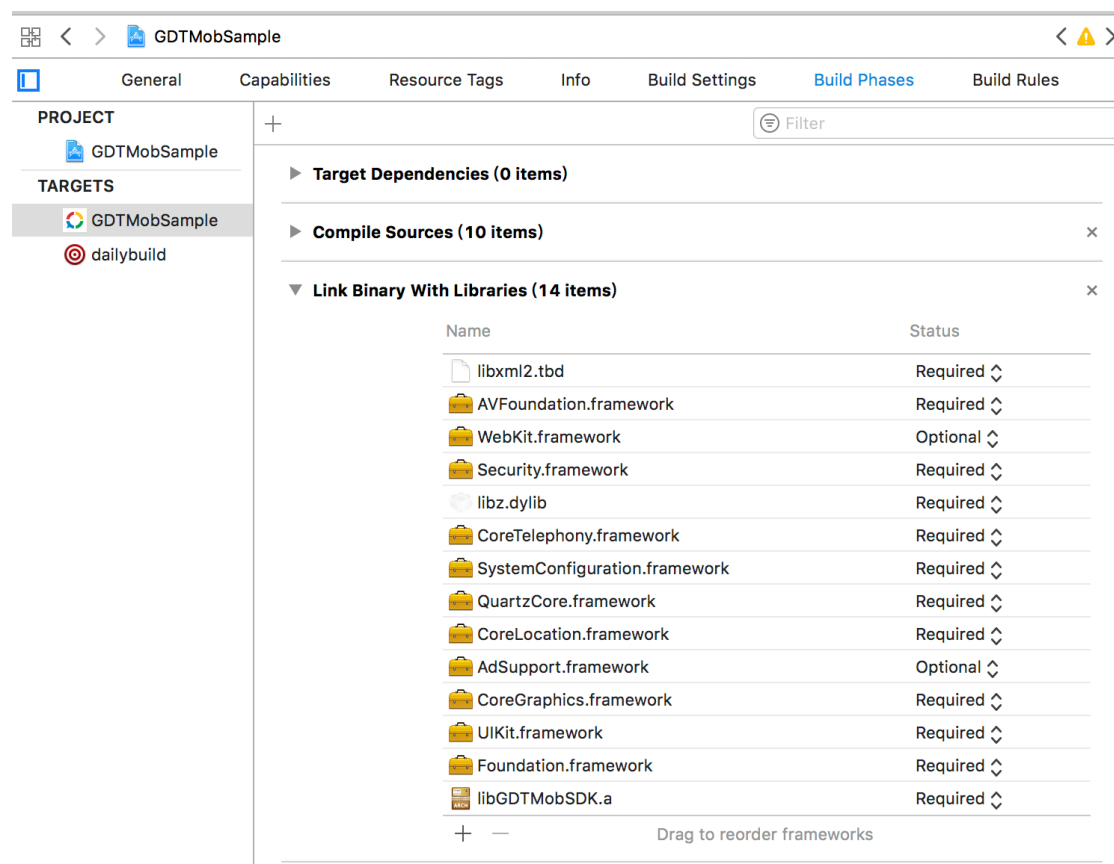




4.3. 为工程添加相应的 Frameworks

需要为项目添加如下 Frameworks

- AdSupport.framework
- CoreLocation.framework
- QuartzCore.framework
- SystemConfiguration.framework
- CoreTelephony.framework
- libz.dylib 或 libz.tbd
- Security.framework
- StoreKit.framework
- WebKit.framework (Optional)
- AVFoundation.framework
- libxml2.tbd



5. 接入代码

5.1. Banner 广告接入代码

（详细内容请参考压缩包中的 Sample 项目）

1. 在需要导入广告的 ViewController 头文件中导入头文件并声明实例：

```
#import <UIKit/UIKit.h>
#import "GDTMobBannerView.h" //导入GDTMobBannerView.h头文件

@interface BannerViewController :
    UIViewController<GDTMobBannerViewDelegate>
{
    GDTMobBannerView *_bannerView; //声明一个GDTMobBannerView的实例
}

@end
```

2. 在 ViewController 的实现文件中初始化并加载广告：

```

@implementation BannerViewController

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
    self = [super initWithNibName:@"BannerView" bundle:nibBundleOrNil];
    if (self) {
        /*
         * 创建Banner广告View
         *
         * banner条的宽度开发者可以进行手动设置，用以满足开发场景需求或是适配最新版本的iphone
         * banner条的高度广点通强烈建议开发者采用推荐的高度，否则显示效果会有影响
         * 广点通提供3种尺寸供开发者在不同设备上使用，这里以320*50为例
         */
        _bannerView = [[GDTMobBannerView alloc] initWithFrame:CGRectMake(0, 0,
                                                                           GDTMOB_AD_SUGGEST_SIZE_320x50.width,
                                                                           GDTMOB_AD_SUGGEST_SIZE_320x50.height)
                       appkey:@"100720253"
                       placementId:@"9079537207574943610"];
    }

    return self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    _bannerView.delegate = self; // 设置Delegate
    _bannerView.currentViewController = self; //设置当前的ViewController
    _bannerView.interval = 30; // 【可选】设置广告轮播时间；范围为30~120秒，0表示不轮播
    _bannerView.isGpsOn = NO; // 【可选】开启GPS定位；默认关闭
    _bannerView.showCloseBtn = YES; // 【可选】展示关闭按钮；默认显示
    _bannerView.isAnimationOn = YES; // 【可选】开启banner轮播和展现时的动画效果；默认开启
    [self.view addSubview:_bannerView]; //添加到当前的view中
    [_bannerView loadAdAndShow]; //加载广告并展示
}

```

备注：

- 1) 在释放 bannerView 之前，建议在先将 bannerView 对象的 delegate 和 currentViewController 属性置为空，然后再释放 bannerView。

2) 开启 GPS 定位可以获取地理位置信息，提高广告填充率，增加收益，但是如果与本身应用功能相矛盾也存在被苹果审核拒绝的风险。

3. 实现 GDTMobBannerViewDelegate 协议：【可选】

您可以通过实现 GDTMobBannerViewDelegate 中定义的方法，来跟踪广告生命周期中的各个阶段，这些方法定义如下：

```
// 请求广告条数据成功后调用
- (void)bannerViewDidReceived;
// 请求广告条数据失败后调用
- (void)bannerViewFailToReceived:(NSError *)error;
// 应用进入后台时调用
- (void)bannerViewWillLeaveApplication;
// 广告条曝光回调
- (void)bannerViewWillExposure
// 广告条点击回调
- (void)bannerViewClicked
// banner条被用户关闭时调用
- (void)bannerViewWillClose
```

5.2. 插屏广告接入代码

（详细内容请参考压缩包中的 Sample 项目）

1. 在想要导入广告的 ViewController 头文件中导入头文件并声明实例：

```
#import <UIKit/UIKit.h>
#import "GDTMobInterstitial.h"

@interface InterstitialViewController :
    UIViewController<GDTMobInterstitialDelegate>
{
    GDTMobInterstitial *_interstitialObj;
}
```

2. 在 ViewController 的实现文件中初始化并预加载广告，之后在合适的时机进行展现：

```
// 在适当的时候，初始化并调用loadAd方法进行预加载
- (IBAction)loadAd:(id)sender {
    _interstitialObj = [[GDTMobInterstitial alloc]
initWithAppkey:@"2211675583"

    placementId:@"2954362461573316031"];
    _interstitialObj.delegate = self; //设置委托
    _interstitialObj.isGpsOn = NO;    //【可选】设置GPS开关
    //预加载广告
    [_interstitialObj loadAd];
}

/**
 * 在适当的时候，调用presentFromRootViewController来展现插屏广告
 */
- (IBAction)showAd:(id)sender {
    UIViewController *vc = [[[UIApplication sharedApplication] keyWindow]
rootViewController];
```

备注：

- 1) 在一次展现完成后，需要再次调用 loadAd 方法来加载新的广告，用于下次展现。
- 2) 关于释放，建议在释放 ViewController 之前将_interstitialObj 的 delegate 属性设置为 nil
3. 实现 GDTMobInterstitialDelegate 方法：【可选】

您可以通过实现 GDTMobInterstitialDelegate 中定义的方法，来跟踪插屏广告生命周期中的各个阶段，这些方法定义如下：

```
// 广告预加载成功回调
- (void)interstitialSuccessToLoadAd:(GDTMobInterstitial *)interstitial;
// 广告预加载失败回调
- (void)interstitialFailToLoadAd:(GDTMobInterstitial *)interstitial
error:(NSError *)error;
// 插屏广告将要展示回调
- (void)interstitialWillPresentScreen:(GDTMobInterstitial *)interstitial;
// 插屏广告视图展示成功回调
- (void)interstitialDidPresentScreen:(GDTMobInterstitial *)interstitial;
// 插屏广告展示结束回调
- (void)interstitialDidDismissScreen:(GDTMobInterstitial *)interstitial;
// 应用进入后台时回调
- (void)interstitialApplicationWillEnterBackground:(GDTMobInterstitial
*)interstitial;
// 插屏广告曝光时回调
- (void) interstitialWillExposure:(GDTMobInterstitial *)interstitial;
```

5.3. 开屏广告接入代码

（详细内容请参考压缩包中的 Sample 项目）

1. 在 AppDelegate 头文件中导入头文件并声明实例：

```
#import "GDTSplashAd.h"

@interface GDTAppDelegate : UIResponder
<UIApplicationDelegate, GDTSplashAdDelegate>

@property (strong, nonatomic) GDTSplashAd *splash;
@property (retain, nonatomic) UIView *bottomView;

@end
```

2. 在 AppDelegate 的实现文件中初始化并加载广告数据，开屏广告目前支持全屏开屏和半屏开屏广告两种形式，其中半屏开屏广告支持开发者自定义设置开屏底部的界面，用以展示应用 Logo 等。嵌入代码如下：

```

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    //App自有逻辑部分

    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    self.viewController = [[ViewController alloc] initWithNibName:@"View" bundle:nil];
    GDTNavigationController *nav = [[GDTNavigationController alloc]
initWithRootViewController:self.viewController];
    self.window.rootViewController = nav;
    [self.window makeKeyAndVisible];

    //开屏广告初始化并展示代码
    GDTSplashAd *splash = [[GDTSplashAd alloc] initWithAppkey:@"3630"
placementId:@"88633644363093"];

    splash.delegate = self; //设置代理
    //根据iPhone设备不同设置不同背景图
    if ([[UIScreen mainScreen] bounds].size.height >= 568.0f) {
        splash.backgroundColor = [UIColor colorWithPatternImage:[UIImage
imageNamed:@"LaunchImage-568h"]];
    } else {
        splash.backgroundColor = [UIColor colorWithPatternImage:[UIImage
imageNamed:@"LaunchImage"]];
    }

    splash.fetchDelay = 3; //开发者可以设置开屏拉取时间，超时则放弃展示
    // [ 可选 ] 拉取并展示全屏开屏广告
    // [splashAd loadAdAndShowInWindow:self.window];

    //设置开屏底部自定义LogoView，展示半屏开屏广告
    _bottomView = [[UIView alloc] initWithFrame:CGRectMake(0, 0, [[UIScreen mainScreen]
bounds].size.width, 100)];

    UIImageView *logo = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"SplashBottomLogo"]];
    [_bottomView addSubview:logo];
    logo.center = _bottomView.center;
    bottomView.backgroundColor = [UIColor whiteColor];
}

```

3. GDTSplashAdDelegate 提供拉取，展示，点击广告的相应状态回调，供开发者使用。

```
@protocol GDTSplashAdDelegate <NSObject>

// 开屏广告成功展示
- (void) splashAdSuccessPresentScreen: (GDTSplashAd *) splashAd;

// 开屏广告展示失败
- (void) splashAdFailToPresent: (GDTSplashAd *) splashAd withError: (NSError *) error;

// 应用进入后台时回调
- (void) splashAdApplicationWillEnterBackground: (GDTSplashAd *) splashAd;

// 开屏广告点击回调
- (void) splashAdClicked: (GDTSplashAd *) splashAd;

// 开屏广告关闭回调
- (void) splashAdClosed: (GDTSplashAd *) splashAd;

@end
```

备注：目前广点通移动联盟开屏广告只针对 iPhone 设备在垂直方向上展示。

5.4. 原生广告（模板方式）接入代码

原生模板广告是腾讯广告联盟推出的一种自动化展现的原生广告。原生模板广告简化了原生广告的接入流程，使得接入原生广告更加便捷。

注意：以下接入代码示例和接口均只适用于原生模板广告位。误用其他广告位 ID 使用原生模板广告的接口，则会导致无广告返回。

您可以这样查看当前广告位是否是原生模板广告位：登录联盟开发者平台，在“修改广告位”界面（【广告位管理】-【广告位列表】，点击对应广告位的【修改】操作），如果广告位类型为“原生广告”，且“样式”栏显示为“上图下文”“上文下图”等选项，则表明当前广告位是原生模板广告位。

如果您需要使用开发者自渲染的老接口，请您参考 5.5 原生广告（自渲染方式）。

（详细内容请参考压缩包中的 Sample 项目）

1. 在开发者自己 ViewController 中,导入头文件，例如：Sample 中的 NativeExpressAdViewController，如下：

```
@interface NativeExpressAdViewController () <GDTNativeExpressAdDelegete>

// 用于请求原生模板广告，注意：不要在广告打开期间释放！
@property (nonatomic, retain) GDTNativeExpressAd *nativeExpressAd;

// 返回的原生模板广告数组
@property (nonatomic, retain) NSArray *expressAdViews;
```

2. 初始化 nativeExpressAd，获取广告，并设置回调。

```

self.nativeExpressAd = [[GDTNativeExpressAd alloc] initWithAppkey:@"1105344611"
placementId:@"4090404001146426" adSize:CGSizeMake(300, 300)];

self.nativeExpressAd.delegate = self;

// 例如：拉取 5 条广告
[self.nativeExpressAd loadAd:5];

```

```

// 取一个 GDTNativeExpressAdView
GDTNativeExpressAdView *expressView = [self.expressAdViews objectAtIndex:0];
// 设置 frame，开发者自己设置，参考广告位模板配置
expressView.frame = CGRectMake(0, 0, width,height);
expressView.controller = rootViewController;

[expressView render];

```

3. 在 nativeExpressAd 的回调中，检测广告回调状态。成功之后，会返回 view 数组；开发者通过取得的 view，对 view 调用 render 方法来渲染，如下：

4. 原生模板广告还提供了丰富的回调接口，包括曝光、点击、广告打开、广告关闭的回调等，详细接口参考 sample 代码。

5.5. 原生广告（自渲染方式）接入代码

注意：以下接入代码示例和接口均只适用于原生自渲染广告位。误用其他广告位 ID 使用原生自渲染广告的接口，将会导致无广告返回。

您可以这样查看当前广告位是否是原生自渲染广告位：登录联盟开发者平台，在“修改广告位”界面（【广告位管理】-【广告位列表】，点击对应广告位的【修改】操作），如果广告位类型为“原生广告”，且“样式”栏显示为“样式 1”“样式 2”等选项，则表明当前广告位是原生自渲染广告位，您可以使用此类接口；如果“样式”栏显示“上图下文”“上文下图”等选项，则表明当前广告位是原生模板广告位，请您使用 5.4 原生广告（模板方式）所述的接口获取广告，或者更换成原生自渲染广告位再使用此接口。

（详细内容请参考压缩包中的 Sample 项目）

1. 在想要导入广告的 ViewController 头文件中导入头文件并声明实例：

```

#import "GDTNativeAd.h"

@interface NativeViewController : UIViewController<GDTNativeAdDelegate>
{
    GDTNativeAd *_nativeAd;           //原生广告实例
    NSArray *_data;                   //原生广告数据数组
    GDTNativeAdData *_currentAd;      //当前展示的原生广告数据对象
    UIView *_adView;                  //当前展示的原生广告界
}

@end

```


2. 在 ViewController 的实现文件中初始化并加载广告数据：

```
- (void)viewDidLoad {
    [super viewDidLoad];
    /*
     * 创建原生广告
     *
     * 本原生广告位ID在联盟系统中创建时勾选的详情图尺寸为1280*720，开发者可以根据自己应用的需要
     * 创建对应的尺寸规格ID
     *
     * 这里详情图以1280*720为例
     */

    _nativeAd = [[GDTNativeAd alloc] initWithAppkey:@"appkey"
placementId:@"pid"];
    _nativeAd.controller = self;
    _nativeAd.delegate = self;
}

- (IBAction)loadAd:(id)sender {
    /*
     * 拉取广告，传入参数为拉取个数。
     * 发起拉取广告请求，在获得广告数据后回调delegate
     */
}
```

3. 在 delegate 回调方法中获取原生广告数据并存储。

```
/*在Delegate回调方法中获取广告数据*/
- (void)nativeAdSuccessToLoad:(NSArray *)nativeAddataArray
{
    /*广告数据拉取成功，存储并展示*/
    _data = nativeAddataArray;
}
```

4. 在合适的时机渲染广告界面并进行展示，目前 GDTNativeAdData 对象中 properties 内容字典包含如下键：

| 键 | 描述 |
|-----------------------------|------------|
| GDTNativeAdDataKeyTitle | 广告标题 |
| GDTNativeAdDataKeyDesc | 广告描述 |
| GDTNativeAdDataKeyIconUrl | 广告图标 URL |
| GDTNativeAdDataKeyImgUrl | 广告详情图 URL |
| GDTNativeAdDataKeyAppRating | 应用类广告的星级数据 |
| GDTNativeAdDataKeyAppPrice | 应用类广告的价格数据 |

GDTNativeAdDataKeyImgList

三小图广告的图片集合

开发者在获取相应数据后按照自身需求渲染广告界面嵌入应用中，需要注意渲染完毕曝光给最终用户时需调用 GDTNativeAd -> attach:toView:方法告知广点通广告已经渲染完毕并曝光。

同时，开发者还需要注意在渲染时给广告界面添加点击事件响应，并在事件中调用 SDK 的点击方法。参考代码如下：（详细代码可参考示例工程）

```
/*渲染广告界面，注册点击事件，上报渲染完毕操作并曝光*/
- (IBAction)attach:(id)sender {
    /*选择展示广告*/
    _currentAd = [_data objectAtIndex:0];
    /*开始渲染广告界面，这里以标题为例*/
    _adView = [[UIView alloc] initWithFrame:CGRectMake(320, 20, 320, 250)];
    UILabel *txt = [[UILabel alloc] initWithFrame:CGRectMake(80, 5, 100, 35)];
    /*从字典中通过相应字段获取数据*/
    txt.text = [_currentAd.properties objectForKey:GDTNativeAdDataKeyTitle];
    /*注册点击事件*/
    UITapGestureRecognizer *tap = [[UITapGestureRecognizer alloc]
initWithTarget:self action:@selector(viewTapped:)];
    [_adView addGestureRecognizer:tap];
    /*广告数据渲染完毕，即将展示时需调用AttachAd方法 */
    [_nativeAd attachAd:_currentAd toView:_adView];
    [_adView addSubview:txt];
}
- (void)viewTapped:(UITapGestureRecognizer *)gr {
    [_nativeAd clickAd:_currentAd]; /*点击发生，调用点击接口*/
}
```

5. GDTNativeAdDelegate 提供拉取广告，点击广告的相应状态回调，供开发者使用。

```
@protocol GDTNativeAdDelegate <NSObject>
// 原生广告加载广告数据成功回调，返回为GDTNativeAdData对象的数组
- (void)nativeAdSuccessToLoad:(NSArray *)nativeAddataArray;
// 原生广告加载广告数据失败回调
- (void)nativeAdFailToLoad:(NSError *)error;

@optional
// 原生广告点击之后将要展示内嵌浏览器或应用内AppStore回调
- (void)nativeAdWillPresentScreen;
// 原生广告点击之后应用进入后台时回调
- (void)nativeAdApplicationWillEnterBackground;
```

备注：

1. 建议在释放原生广告 GDTNativeAd 对象之前将 GDTNativeAd 对象的 delegate 属性设置为 nil
2. 当用户点击广告弹出内置目标页时，请勿释放 GDTNativeAd 实例。
3. 广告从拉取到曝光超过 45 分钟，将作为无效曝光。
4. 广告从拉取到点击超过 90 分钟，将作为无效点击，不会进行计费。
5. 广告数据渲染完毕，即将展示时需调用 AttachAd 方法，否则将不会产生曝光记录，也无法进行计费。

6. 问题排查

如果根据正常的注册流程仍然无法在嵌入广点通 SDK 的 app 中看到广告，可以在各个广告形式的 delegate 失败回调方法中输出错误信息。

广点通在打印广告关键信息时会带上 **[GDTMob::GDTMob]** 的标记，其他带有 GDTMob 标记的 log 与广告信息无关。

| errorCode | 描述 |
|-----------|--|
| 4001 | 初始化错误, 包括广告位为空、AppKey 为空、ViewController 为空 |
| 4003 | 广告位错误 |
| 4006 | 广告未曝光 |
| 4007 | 设备不支持 |
| 4008 | 设备方向不支持 |
| 4009 | 开屏跳过按钮定义非法 |
| 4010 | 开屏 bottomView 设置非法 |
| 4011 | 请求广告超时 |
| 4012 | 系统不支持, 原生视频模板广告只支持 iOS 9 及以上系统。 |
| 5001 | 后台数据错误 |
| 5004 | 没有广告 |
| 5005 | 广告请求量或者消耗等超过日限额, 请第二天再请求广告 |
| 5006 | 包名校验非法 |
| 5009 | 广告请求量或者消耗等超过小时限额, 请一小时后再请求广告 |
| 5010 | 广告样式校验失败, 请检查广告位与接口使用是否一致 |
| 6000 | 未知错误, 联系广点通商务同事协助排查 |

7. 注意事项

1. 在以下情况下使用 Banner 广告请将 interval 刷新频率设为 0，即不自动刷新。

- a) 使用广告聚合平台控制刷新频率时。
- b) 使用代码定时新建 GDTMobBannerView 来控制刷新频率时。

2. 苹果公司在 iOS9 中升级了应用网络通信安全策略，默认推荐开发者使用 HTTPS 协议来进行网络通信，并限制 HTTP 协议的请求。为了避免出现无法拉取到广告的情况，我们推荐开发者在 info.plist 文件中增加如下配置来实现广告的网络访问：（信任 HTTP 请求）

若开发者有使用 HTTPS 进行应用内网络访问的场景需要，可以调用广点通提供的接口，我们会使用 HTTPS 资源进行广告请求和资源渲染。如下：

```
@implementation GDTAppDelegate
@synthesize viewController;
@synthesize window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    // Override point for customization after application launch.
    self.viewController = [[ViewController alloc] initWithNibName:@"View" bundle:nil];
    GDTNavigationController *nav = [[GDTNavigationController alloc] initWithRootViewController:self.viewController];
    nav.navigationBar.barStyle = UIBarStyleBlackOpaque;
    nav.navigationBar.topItem.title = @"广告形式";

    self.window.rootViewController = nav;
    [self.window makeKeyAndVisible];

    [GDTSDKConfig setHttpsOn];

    return YES;
}
```