iOS PandaADSDK

开发者手册

PandaAdSDK支持如下功能:
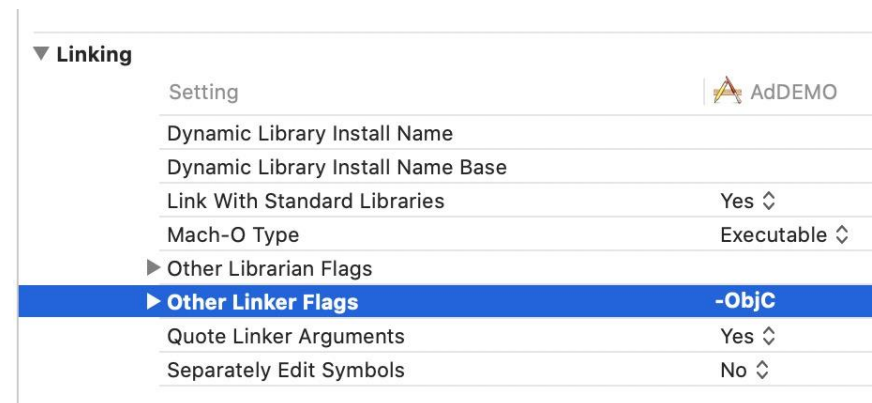
功能	详情
开屏广告	开屏广告
模板信息流广告	模板信息流广告
激励视频	激励视频
资讯功能	资讯功能
营销中心	营销中心
优惠券商城	优惠券商城

SDK接入流程

将Demo中lib文件夹下的所有文件拖入到需要添加的项目当中
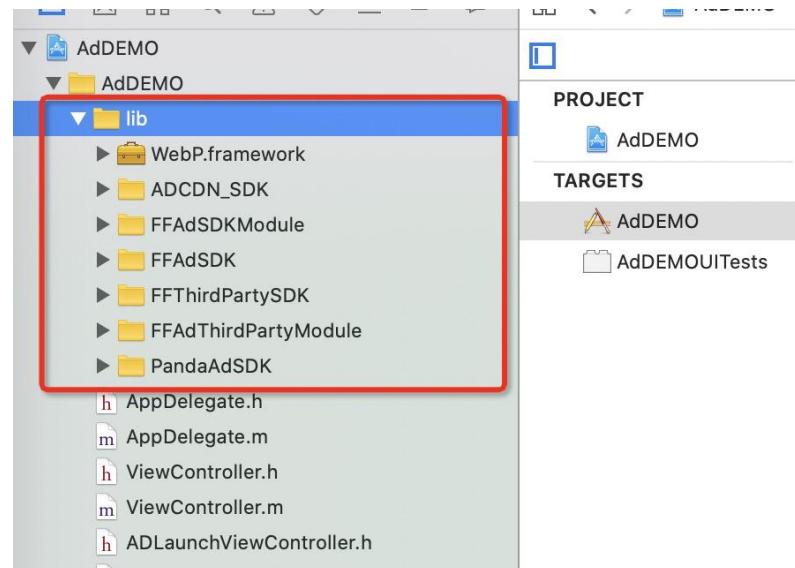
在项目设置中设置 Other　Linker　Flags 的值，并添加"-ObjC"标志；

Xcode info.plist 相关配置

ps:需加入定位配置信息，我们会精准定制给推荐更有效的广告

在 info.plist 中添加这两个即可：

Privacy - Location Always and When In Use Usage Description

Privacy - Location When In Use Usage Description

在info.plist中增加权限 NSAppTransportSecurity 如下 权限申请 中的内容

Application requires iPhone environ...	⌄	Boolean	YES
▼ App Transport Security Settings	⌃⌄	Dictionary	(1 item)
Allow Arbitrary Loads	⌃⌄	Boolean	YES
Privacy - Location Always and Wh...	⌃⌄	String	需加入定位配置信息，我们会精准定制给推荐更有效的广告
Privacy - Location When In Use Us...	⌃⌄	String	需加入定位配置信息，我们会精准定制给推荐更有效的广告

在工程中添加 SDK 以及各广告依赖的 framework；

Accelerate.framework
libsqlite3.tbd
libc++.tbd
MobileCoreServices.framework
AVFoundation.framework
AdSupport.framework --- Optional
CoreLocation.framework
CoreMedia.framework
CoreMotion.framework
CoreTelephony.framework
ImageIO.framework
MediaPlayer.framework
QuartzCore.framework
Foundation.framework
Security.framework
StoreKit.framework --- Optional
SystemConfiguration.framework
UIKit.framework
WebKit.framework
libxml2.tbd
libz.tbd
libz.1.2.5.tbd
libbz2.tbd
libresolv.tbd
libiconv.tbd

接入代码

程序启动初始化SDK

SDK初始化配置，在AppDelegate.m中导入SDK的头文件：#import

<PandaAdSDK/PandaConfigManager.h>

，在app程序的启动函数didFinishLaunchingWithOptions中初始化SDK 提示：appId由熊猫提供，通常为项目名小写全拼

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    // Override point for customization after application launch.

    self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
    [self.window makeKeyAndVisible];

    [PandaConfigManager initSDKWithAppId:@"xiongmaochuxing"];
```

旧版本用户更新请注意(新接入项目可以忽略)

1.先将旧版本SDK全部删除后，再接入新SDK。防止出现引用到旧版本SDK的情况。

2.对于旧版本用到下方代码统计的项目，新版本不再需要调用此方法

```
    //统计启动次数
    //AppId:项目名称（全拼小写字母）
    [PBFFStatisticsManager initWithAppId:@"pandabusTest" type:0 completed:^(BOOL success
NSError * _Nullable error) {


    }];
```

如果之前传的appId与新分配的appId（应用ID）相同，正常调用PandaConfigManager的初始化方法即可

```
/**
 *  初始化
 *  熊猫出行广告初始化
 *  appId - 应用 ID
 */
+ (void)initSDKWithAppId:(NSString *)appId;
```

如果之前传的appId与新分配的appId（应用ID）不同，需要调用PandaConfigManager

```
/**
 *  初始化
 *  熊猫出行广告初始化
 *  appId - 应用 ID
 *  新接入此SDK项目忽略此方法，用上面的初始化方法
 */
+ (void)initSDKWithAppId:(NSString *)appId
                cityCode:(NSString *)cityCode;
```
方法，appId传新分配的id，cityCode传旧的appId

（即旧版本PBFFStatisticsManager统计方法传的appId）

PandaConfigManager中的两个初始化方法二选其一调用即可，不需要都调用。

开屏广告

Appdelegate添加PandaSplashAdManager属性，注意：此属性需要强引用，否则会出现代理无法回调的问题

```objc
@interface AppDelegate ()<PandaSplashAdManagerDelegate>

@property (nonatomic, strong, PandaSplashAdManager * splashManager;

@end
```

初始化并加载开屏广告

```objc
#pragma mark - 加载开屏广告
-(void)loadSplashAd {
    CGRect frame = [UIScreen mainScreen].bounds;
    // bottomView - [可选]、兜底图、防止开屏广告材料还没拉取到、会有白屏展示、建议：兜底可以设置为启
动图。
    UIView *bottomView = [[UIView alloc] initWithFrame:frame];
    UIImageView * bottomImageView = [[UIImageView alloc]initWithFrame:frame];
    bottomImageView.image = [UIImage imageNamed:@"bottomView_image"];
    [bottomView addSubview:bottomImageView];

    //logoView -广告发起请求并展示在Window中，同时在屏幕底部设置应用自身的Logo页面或是自定义View
    CGFloat height = [UIScreen mainScreen].bounds.size.width/(750.0/200.0);
    UIView *logoView = [[UIView alloc]initWithFrame:CGRectMake(0, [UIScreen
mainScreen].bounds.size.height - height, [UIScreen mainScreen].bounds.size.width,
height)];
    logoView.backgroundColor = [UIColor whiteColor];
    UIImageView *logoImageView = [[UIImageView alloc]initWithImage:[UIImage
imageNamed:@"logo_image"]];

    CGFloat logoImgHeight = logoView.bounds.size.height;
    CGFloat logoImgWidth = logoView.bounds.size.width;

    logoImageView.frame = CGRectMake((logoView.frame.size.width - logoImgWidth)/2.0,
(logoView.frame.size.height - logoImgHeight)/2.0, logoImgWidth, logoImgHeight);
    [logoView addSubview:logoImageView];

    self.splashManager = [PandaSplashAdManager defaultClient];
    [self.splashManager initSplashAdWithPlcId:@"" TotalFetchDelay:5 window:self.window
bottomView:bottomView logoView:logoView delegate:self
rootvc:self.window.rootViewController];

}
```

实现开屏广告代理方法

```objc
/**
 *  开屏广告成功加载
 */
- (void)PBSplashAdDidLoadData {
```

```objc
    NSLog(@"PandaAdLOG%s---%@",__FUNCTION__,@"开屏广告成功展示");
}


/**
 *  开屏广告展示失败
 */
- (void)PBSplashAdFailToLoad:(NSError *)error {
    NSLog(@"PandaAdLOG%s---%@ error:%@",__FUNCTION__,@"开屏广告展示失败",error);
}


/**
 *  开屏广告曝光回调
 */
- (void)PBSplashAdExposured {
    NSLog(@"PandaAdLOG%s---%@",__FUNCTION__,@"开屏广告曝光回调");
}


/**
 *  开屏广告点击回调
 */
- (void)PBSplashAdClicked {
    NSLog(@"PandaAdLOG%s---%@",__FUNCTION__,@"开屏广告点击回调");
}


/**
 *  开屏广告关闭回调
 */
- (void)PBSplashAdClosed {
    NSLog(@"PandaAdLOG%s---%@",__FUNCTION__,@"开屏广告关闭回调");
}


/**
 *  落地页关闭
 */
- (void)PBSplashAdDissmissDetail {
    NSLog(@"PandaAdLOG%s---%@",__FUNCTION__,@"开屏详情页关闭回调");
}
```

模板信息流广告

初始化模板信息流广告

```objc
self.manager = [[PandaNativeExpressAdManager alloc]init];
```

```
[self.manager initNativeExpressAdWithPlcId:@"" rootvc:self delegate:self
adSize:self.adView.bounds.size];
```

实现模板信息流广告代理方法
在渲染成功的代理中，将回调的view添加到视图上即可

```objc
/**
 *  加载成功
 */
- (void)PandaAdNativeExpressAdSuccessToLoad {
    NSLog(@"加载成功");
}
/**
 *  加载失败
 *  广告拉取失败，禁止多次重试请求广告，避免求量消耗过大，导致填充率过低，影响系统对您流量的评价从而影
响变现效果，得不到广告收益。
 */
- (void)PandaAdNativeExpressAdDidFailWithError:(id)error {
    NSLog(@"加载失败");
}
/**
 *  渲染广告成功
 */
- (void)PandaAdNativeExpressAdRenderSuccessWithView:(UIView *)view
manager:(PandaNativeExpressAdManager *)manager {
    NSLog(@"渲染广告成功");
    [self.adView addSubview:view];
}
/**
 *  渲染广告失败
 */
- (void)PandaAdNativeExpressAdRenderFailWithError:(id)error {
    NSLog(@"渲染广告失败");
}
/**
 *  点击广告
 */
- (void)PandaAdNativeExpressAdDidClick {
    NSLog(@"点击广告");
}
/**
 *  曝光回调
 */
- (void)PandaAdNativeExpressAdDidBecomeVisible {
```

```
    NSLog(@"曝光回调");
}
/**
 *  关闭广告回调
 */
- (void)PandaAdNativeExpressAdDidClose {
    NSLog(@"关闭广告回调");
}
```

激励视频广告

初始化激励视频广告
```
self.manager = [[PandaRewardVideoAdManager alloc]init];
[self.manager initRewardVideoAdWithPlcId:@"" delegate:self rootvc:self];
```

实现激励视频广告代理方法
```
/**
 *  加载成功
 */
- (void)PandaAdRewardVideoAdDidLoad {
    NSLog(@"加载成功");
}
/**
 *  加载失败
 */
- (void)PandaAdRewardVideoAdDidFailWithError:(id)error {
    NSLog(@"加载失败");
}
/**
 *  下载成功
 */
- (void)PandaAdRewardVideoAdDidDownLoadVideo {
    NSLog(@"下载成功");
}
/**
 *  点击广告
 */
- (void)PandaAdRewardVideoAdDidClick {
    NSLog(@"点击广告");
}
/**
 *  曝光回调
```

```
*/
- (void)PandaAdRewardVideoAdDidBecomeVisible {
    NSLog(@"曝光回调");
}
/**
 *  视频播放完成
 */
- (void)PandaAdRewardVideoAdDidPlayFinishWithError:(id)error {
    NSLog(@"视频播放完成");
}
/**
 * 播放完成点击关闭
 */
- (void)PandaAdRewardVideoAdDidClose {
    NSLog(@"播放完成点击关闭");
}
```
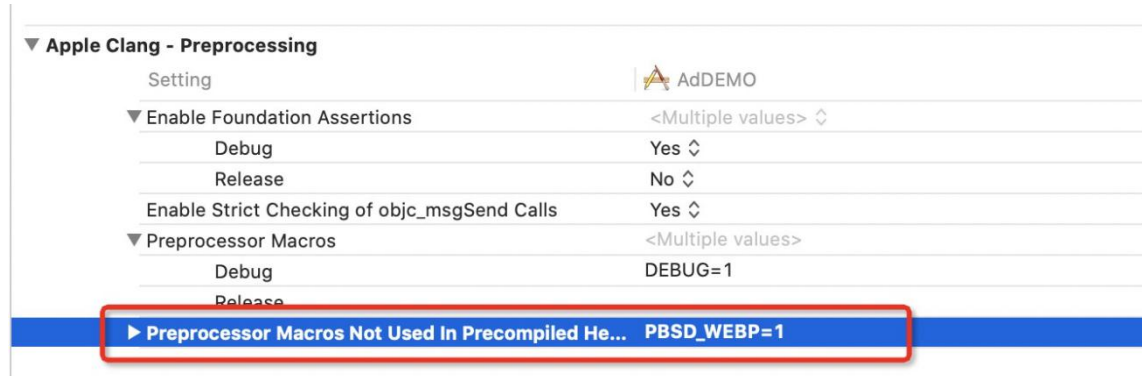
资讯功能

项目配置 设置Build Settings -- Preprocessing , add "PBSD_WEBP=1"

加载资讯控制器
PBAdInfoMainCtrl为资讯viewController，可根据项目需要，加载此页面
初始化方法：- (instancetype)initWithProid:(NSString *)proid; proid为熊猫提供

```
ctrl = [[PBAdInfoMainCtrl alloc] initWithProid:@""];
```

营销中心

1.加载营销中心
在需要加载营销活动的ViewController调用如下方法
参数说明：

```
/**
 *  加载营销活动
 *  AppId - 营销中心appId(注意不是广告appId，字母数字组合，由熊猫提供。)
 *  splashPlcId - 营销中心开屏广告位id，由熊猫提供
 *  rewardVideoPlcId - 营销中心激励视频广告位id，由熊猫提供
 *  userId - app的用户id，用于区分用户的标识
 *  superController - 加载营销活动的VC
 *  marketingShowCode - 用于区分显示活动位置的code，不同位置加载传不同的showcode，熊猫提供
 *  titleColor - 导航栏title以及icon颜色
 *  backColor - 导航栏背景颜色
 */
```

```
#import <PandaAdSDK/PBADMarketingActivityManager.h>
```

```
[PBADMarketingActivityManager initWithAppId:@"" splashPlcId:@"" rewardVideoPlcId:@""
userId:@"" superController:self marketingShowCode:@"" titleColor:[UIColor whiteColor]
backColor:[UIColor blackColor] complete:^(PBADMarketingActivityModel *_Nonnull dataModel
BOOL success, NSError *_Nonnull error) {

    }];
```

2.加载活动列表页面
跳转至营销中心活动列表

```
    PBADAvtivityListViewController *controller = [[PBADAvtivityListViewController alloc]
initWithNibName:@"PBADAvtivityListViewController" bundle:nil];
    controller.appId = @"营销中心Id";
    controller.userId = @"app用户id";
    controller.titleStr = @"优惠活动";
    controller.splashPlcId = @"广告位id";
    controller.backColor = [UIColor whiteColor];
    controller.titleColor = [UIColor blackColor];
    controller.hidesBottomBarWhenPushed = YES;
    [self.navigationController pushViewController:controller animated:YES];
```

优惠券商城

在项目info.plist中LSApplicationQueriesSchemes下添加下面字段

```
<key>LSApplicationQueriesSchemes</key>
<array>
<string>didipasnger</string>
<string>alipayshare</string>
```

```xml
<string>alipay</string>
<string>weixin</string>
<string>wechat</string>
<string>weibosdk2.5</string>
<string>weibosdk</string>
<string>sinaweibohdsso</string>
<string>sinaweibosso</string>
<string>sinaweibohd</string>
<string>sinaweibo</string>
<string>openApp.jdMobile  </string>
<string>taobao</string>
<string>pinduoduo</string>
<string>tbopen</string>
<string>tmall</string>
<string>eleme</string>
<string>vipshop</string>
<string>itaobao</string>
<string>suning</string>
<string>alipayqr</string>
<string>jdlogin</string>
<string>weixinULAPI</string>
</array>
```

加载优惠券商城

```objc
ctrl = [[PBShopWebViewController alloc] initWithTitle:@"" iconColor:[UIColor blackColor]
webViewHeight:[UIScreen mainScreen].bounds.size.height];
[self.navigationController pushViewController:ctrl animated:YES];
```

PBShopWebViewController 为SDK提供的webView容器，便于开发者集成开发。

开发者也可以自行创建ViewController加载商城url实现自定样式以及业务：
url获取方式：
PandaConfigManager中

```objc
/**
 *  获取优惠券商城h5 Url（初始化后调用）
 */
+ (NSString *)getShopUrl;
```

WKWebView代理方法需实现:

```objc
- (void)webView:(WKWebView *)webView
decidePolicyForNavigationAction:(WKNavigationAction *)navigationAction
decisionHandler:(void (^)(WKNavigationActionPolicy))decisionHandler {
    NSURL *url = navigationAction.request.URL;
```

```objc
    if (navigationAction.targetFrame == nil) {
        [webView loadRequest:navigationAction.request];
    }
    if ([url.scheme hasPrefix:@"tel"]) {
        if ([[UIApplication sharedApplication] canOpenURL:url]) {
            [[UIApplication sharedApplication] openURL:url];
            decisionHandler(WKNavigationActionPolicyCancel);
            return;
        }
        decisionHandler(WKNavigationActionPolicyAllow);
        return;
    }
    if (!IsStrEmpty(url.scheme) && ![url.scheme hasPrefix:@"http"] && ![url.scheme
hasPrefix:@"about"]) {
        if ([[UIApplication sharedApplication] canOpenURL:url]) {
            if ([[UIApplication sharedApplication] canOpenURL:url]) {
                [[UIApplication sharedApplication] openURL:url];
                decisionHandler(WKNavigationActionPolicyCancel);
                return;
            }
        }
        decisionHandler(WKNavigationActionPolicyAllow);
        return;
    }
    decisionHandler(WKNavigationActionPolicyAllow);
}
```

其他

关于提交AppStore 隐私信息，请参照附1文档。
关于App Tracking Transparency 权限，请参照附2文档（可选）。