LYJAdSDK iOS SDK 接入文档

本文档适用于 iOS App 接入 `LYJAdSDK` 广告聚合能力。示例中的 `appId`、广告位 ID 仅用于 Demo，正式环境请替换为 LYJAdSDK 后台分配的参数。

1. 环境要求

- iOS 12.0 及以上
- Xcode 12 及以上
- Objective-C / Swift 工程均可接入
- 工程需开启 ARC
- SDK 当前 podspec 版本：`6.9.1`

2. SDK 导入

2.1 CocoaPods 接入

在 `Podfile` 中添加：

```
target 'YourApp' do
  use_frameworks!

  pod 'LYJAdSDK', :path => './LYJAdSDK'
end
```

然后执行：

```
pod install
```

当前 `LYJAdSDK.podspec` 会自动引入 SDK 依赖的第三方广告平台：

```
pod 'KSAdSDK', '5.1.20.1'
pod 'GDTMobSDK', '4.15.75'
pod 'Ads-CN', '7.4.0.4'
pod 'BaiduMobAdSDK', '10.032'
```

```
pod 'JADYun', '2.6.8'
pod 'TanxSDK', '3.7.4'
```

2.2 手动接入

如不使用 CocoaPods，需要将以下内容加入主工程：

- `LYJAdSDK.framework`
- `LYJAdSDK.framework/lyjadsdk.bundle`

并在 `Build Settings` / `Other Linker Flags` 中加入：

```
$(inherited) -ObjC
```

同时确认链接以下系统库：

```
AdSupport
AppTrackingTransparency
AudioToolbox
AVFoundation
CoreGraphics
CoreLocation
CoreMedia
CoreMotion
CoreTelephony
Foundation
ImageIO
JavaScriptCore
MobileCoreServices
QuartzCore
SafariServices
Security
StoreKit
SystemConfiguration
UIKit
WebKit
AdServices
libc++.tbd
libresolv.tbd
libsqlite3.tbd
```

```
libxml2.tbd
libz.tbd
```

3. 工程配置

3.1 导入头文件

Objective-C：

```
#import <LYJAdSDK/LYJAdSDK.h>
```

Swift 工程可在 Bridging Header 中加入：

```
#import <LYJAdSDK/LYJAdSDK.h>
```

3.2 Info.plist 配置

Demo 中包含以下配置，可按业务实际情况添加：

```
<key>NSUserTrackingUsageDescription</key>
<string>用于向您提供更相关的广告内容</string>

<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
</dict>

<key>LSApplicationQueriesSchemes</key>
<array>
    <string>taobao</string>
    <string>iMeituan</string>
    <string>openapp.jdmobile</string>
    <string>tmall</string>
    <string>tbopen</string>
    <string>pddopen</string>
</array>
```

如 App 使用 IDFA，请在合适时机向用户申请 ATT 授权。

4. SDK 初始化

建议在 `AppDelegate` 的 `application:didFinishLaunchingWithOptions:` 中初始化，广告请求前必须先调用 `config:mediaVersion:`。

```
#import <LYJAdSDK/LYJAdSDK.h>

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionar
    NSLog(@"当前已成功导入的SDK：%@", [LYJAdSDKConfig.supportSDKs componentsJoinedByString:@",

    [LYJAdSDKConfig config:@"100012" mediaVersion:@"1.2"];
    [LYJAdSDKConfig setDebug:YES];

    NSLog(@"LYJAdSDK version = %@", LYJAdSDKConfig.version);
    return YES;
}
```

常用配置：

```
// 限制个性化广告，默认 NO
[LYJAdSDKConfig setLimitPersonalAds:YES];

// 设置媒体侧用户 ID
[LYJAdSDKConfig setCustomUserId:@"your_user_id"];

// 媒体自定义参数
[LYJAdSDKConfig setExtraInfo:@{@"key": @"value"}];

// 当 App 无法获取 IDFA 且媒体侧已合法获取 IDFA 时，可传入自定义 IDFA
[LYJAdSDKConfig setCustomIDFA:@"your_idfa"];
```

如需开启主动反调试防御，必须在初始化前调用：

```
[LYJAdSDKConfig setActiveAntiDebugEnabled:YES];
[LYJAdSDKConfig config:@"100012" mediaVersion:@"1.2"];
```

5. 开屏广告

5.1 创建并加载

```
@interface ViewController () <LYJAdSplashAdDelegate>
@property (nonatomic, strong) LYJAdSplashAd *splashAd;
@end

- (void)loadSplashAd {
    UIWindow *window = UIApplication.sharedApplication.delegate.window;

    UIView *bottomView = [self bottomView]; // 可为空，高度建议不超过屏幕 1/4
    LYJAdSplashAd *splashAd = [[LYJAdSplashAd alloc] initWithPlacementId:@"1024"
                                                                  window:window
                                                              bottomView:bottomView
                                                                 timeout:6];
    splashAd.delegate = self;
    [splashAd loadAd];
    self.splashAd = splashAd;
}
```

5.2 回调处理

```
- (void)lyjAdSplashAdLoadSuccess:(LYJAdSplashAd *)splashAd {
    [splashAd showAd];
}

- (void)lyjAdSplashAdLoadFailed:(LYJAdSplashAd *)splashAd error:(NSError *)error {
    NSLog(@"开屏加载失败：%@", error);
}

- (void)lyjAdSplashAdShowFailed:(LYJAdSplashAd *)splashAd error:(NSError *)error {
    NSLog(@"开屏展示失败：%@", error);
}

- (void)lyjAdSplashAdVisible:(LYJAdSplashAd *)splashAd {
    NSLog(@"开屏已展示");
}
```

```
- (void)lyjAdSplashAdDidClick:(LYJAdSplashAd *)splashAd {
    NSLog(@"开屏点击");
}

- (void)lyjAdSplashAdDidClose:(LYJAdSplashAd *)splashAd {
    NSLog(@"开屏关闭");
}
```

6. 插屏广告

```
@interface ViewController () <LYJAdInterstitialAdDelegate>
@property (nonatomic, strong) LYJAdInterstitialAd *interstitialAd;
@end

- (void)loadInterstitialAd {
    self.interstitialAd = [[LYJAdInterstitialAd alloc] initWithPlacementId:@"1025"
                                                                viewController:self];
    self.interstitialAd.delegate = self;
    [self.interstitialAd loadAd];
}

- (void)lyjAdInterstitialAdLoadSuccess:(LYJAdInterstitialAd *)interstitialAd {
    [interstitialAd showAd];
}

- (void)lyjAdInterstitialAdLoadFailed:(LYJAdInterstitialAd *)interstitialAd error:(NSError
    NSLog(@"插屏加载失败：%@", error);
}

- (void)lyjAdInterstitialAdVisible:(LYJAdInterstitialAd *)interstitialAd {
    NSLog(@"插屏已展示");
}

- (void)lyjAdInterstitialAdDidClick:(LYJAdInterstitialAd *)interstitialAd {
    NSLog(@"插屏点击");
}

- (void)lyjAdInterstitialAdDidClose:(LYJAdInterstitialAd *)interstitialAd {
```

```
    NSLog(@"插屏关闭");
}
```

7. 模板信息流广告

模板信息流由 SDK 渲染广告 View。加载成功后需先调用 `render`，在渲染成功回调中再将 `adObject.view` 加入列表。

7.1 创建并加载

```
@interface FeedViewController () <LYJAdExpressFeedAdDelegate>
@property (nonatomic, strong) LYJAdExpressFeedAd *expressFeedAd;
@property (nonatomic, strong) NSMutableArray *dataArray;
@end

- (void)loadExpressFeedAd {
    CGFloat width = UIScreen.mainScreen.bounds.size.width;
    self.expressFeedAd = [[LYJAdExpressFeedAd alloc] initWithPlacementId:@"117"
                                                                   width:width
                                                          viewController:self];
    self.expressFeedAd.delegate = self;
    [self.expressFeedAd loadAds:1];
}
```

7.2 加载与渲染回调

```
- (void)lyjAdExpressFeedAdLoadSuccess:(LYJAdExpressFeedAd *)expressFeedAd
                           adObjects:(NSArray<LYJAdExpressFeedAdObject *> *)adObjects {
    for (LYJAdExpressFeedAdObject *obj in adObjects) {
        [obj render];
    }
}

- (void)lyjAdExpressFeedAdViewRenderSuccess:(LYJAdExpressFeedAd *)expressFeedAd
                                   adObject:(LYJAdExpressFeedAdObject *)adObject {
    [self.dataArray addObject:adObject];
    [self.tableView reloadData];
}
```

```
- (void)lyjAdExpressFeedAdLoadFailed:(LYJAdExpressFeedAd *)expressFeedAd error:(NSError *)e
    NSLog(@"模板信息流加载失败：%@", error);
}

- (void)lyjAdExpressFeedAdViewRenderFail:(LYJAdExpressFeedAd *)expressFeedAd
                                adObject:(LYJAdExpressFeedAdObject *)adObject {
    NSLog(@"模板信息流渲染失败");
}
```

7.3 在 UITableView 中展示

```
- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath
    id data = self.dataArray[indexPath.row];
    if ([data isKindOfClass:LYJAdExpressFeedAdObject.class]) {
        LYJAdExpressFeedAdObject *adObject = (LYJAdExpressFeedAdObject *)data;
        UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"adCell"];
        if (!cell) {
            cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault reuse
        }

        [[cell.contentView viewWithTag:100] removeFromSuperview];
        adObject.view.tag = 100;
        adObject.view.frame = CGRectMake(0, 0, adObject.view.frame.size.width, adObject.vie
        [cell.contentView addSubview:adObject.view];
        return cell;
    }

    return [tableView dequeueReusableCellWithIdentifier:@"normalCell"];
}

- (CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath:(NSIndexPath *)indexP
    id data = self.dataArray[indexPath.row];
    if ([data isKindOfClass:LYJAdExpressFeedAdObject.class]) {
        LYJAdExpressFeedAdObject *adObject = (LYJAdExpressFeedAdObject *)data;
        return adObject.view.frame.size.height;
    }
    return 50;
}
```

7.4 曝光上报

如果后台配置了百度信息流广告，广告进入可视区域后需要调用 `trackImpression`。多次调用不影响结果。

```
- (void)tableView:(UITableView *)tableView willDisplayCell:(UITableViewCell *)cell
forRowAtIndexPath:(NSIndexPath *)indexPath {
    id data = self.dataArray[indexPath.row];
    if ([data isKindOfClass:LYJAdExpressFeedAdObject.class]) {
        [(LYJAdExpressFeedAdObject *)data trackImpression];
    }
}
```

广告被关闭时，需要从列表移除：

```
- (void)lyjAdExpressFeedAdViewDidRemoved:(LYJAdExpressFeedAd *)expressFeedAd
                                adObject:(LYJAdExpressFeedAdObject *)adObject {
    [self.dataArray removeObject:adObject];
    [self.tableView reloadData];
}
```

8. 原生自渲染广告

原生广告由媒体侧自行布局 UI，SDK 提供标题、描述、图片、Logo、视频 View 等物料。

8.1 创建并加载

```
@interface NativeViewController () <LYJNativeAdsManagerDelegate, LYJNativeAdDelegate>
@property (nonatomic, strong) LYJNativeAdsManager *nativeAdsManager;
@property (nonatomic, strong) LYJNativeAd *nativeAd;
@end

- (void)loadNativeAd {
    LYJVideoConfig *config = LYJVideoConfig.new;
    config.userControlEnable = YES;

    self.nativeAdsManager = [[LYJNativeAdsManager alloc] initWithPlacementId:@"501001"
                                                                viewController:self];
    self.nativeAdsManager.delegate = self;
    self.nativeAdsManager.videoConfig = config;
```

```
    [self.nativeAdsManager loadAds:1];
}
```

8.2 加载回调

```
- (void)lyjNativeAdsManagerSuccessToLoad:(LYJNativeAdsManager *)adsManager
                                nativeAds:(NSArray<LYJNativeAd *> *)nativeAdArray {
    for (LYJNativeAd *nativeAd in nativeAdArray) {
        nativeAd.delegate = self;
        [self renderNativeAd:nativeAd];
    }
}

- (void)lyjNativeAdsManager:(LYJNativeAdsManager *)adsManager didFailWithError:(NSError *)e
    NSLog(@"原生广告加载失败：%@", error);
}
```

8.3 渲染物料并注册点击区域

```
- (void)renderNativeAd:(LYJNativeAd *)nativeAd {
    [self.nativeAd unregister];
    self.nativeAd = nativeAd;

    self.titleLabel.text = nativeAd.data.adTitle;
    self.descLabel.text = nativeAd.data.adDescription;

    NSData *imageData = [NSData dataWithContentsOfURL:[NSURL URLWithString:nativeAd.data.ad
    self.imageView.image = [UIImage imageWithData:imageData];

    NSData *logoData = [NSData dataWithContentsOfURL:[NSURL URLWithString:nativeAd.data.sdk
    self.sdkLogoView.image = [UIImage imageWithData:logoData];

    if (nativeAd.data.adMode == LYJNativeAdModeVideo) {
        nativeAd.mediaView.frame = self.imageView.bounds;
        [self.imageView addSubview:nativeAd.mediaView];
        [nativeAd reSizeMediaView];
    }

    [nativeAd registerContainer:self.adContainerView
```

```
            withClickableViews:@[self.adContainerView]];
}
```

可读取的常用物料字段：

```
nativeAd.data.adImage;
nativeAd.data.icon;
nativeAd.data.adLogo;
nativeAd.data.sdkLogo;
nativeAd.data.adTitle;
nativeAd.data.adDescription;
nativeAd.data.adMode;
nativeAd.data.score;
nativeAd.data.commentNum;
nativeAd.data.appSize;
nativeAd.data.videoDuration;
nativeAd.data.appPrice;
nativeAd.data.advertiser;
nativeAd.data.brandName;
```

8.4 曝光与点击回调

如果后台配置了百度信息流广告，广告进入可视区域后需要调用：

```
[nativeAd trackImpression:adView];
```

广告事件回调：

```
- (void)lyjNativeAdDidBecomeVisible:(LYJNativeAd *)nativeAd {
    NSLog(@"原生广告曝光");
}

- (void)lyjNativeAdDidClick:(LYJNativeAd *)nativeAd {
    NSLog(@"原生广告点击");
}
```

页面释放或广告复用前建议解绑：

```
[self.nativeAd unregister];
```

9. 激励视频广告

```
@interface ViewController () <LYJAdRewardVideoAdDelegate>
@property (nonatomic, strong) LYJAdRewardVideoAd *rewardVideoAd;
@end

- (void)loadRewardAd {
    self.rewardVideoAd = [[LYJAdRewardVideoAd alloc] initWithPlacementId:@"test-1229"
                                                                viewController:self];
    self.rewardVideoAd.delegate = self;
    [self.rewardVideoAd loadAd];
}

- (void)lyjAdRewardVideoAdLoadSuccess:(LYJAdRewardVideoAd *)rewardVideoAd {
    if (rewardVideoAd.isAdValid) {
        [rewardVideoAd showAd];
    }
}

- (void)lyjAdRewardVideoAdRewarded:(LYJAdRewardVideoAd *)rewardVideoAd {
    NSLog(@"激励视频发放奖励");
}

- (void)lyjAdRewardVideoAdLoadFailed:(LYJAdRewardVideoAd *)rewardVideoAd error:(NSError *)e
    NSLog(@"激励视频加载失败：%@", error);
}

- (void)lyjAdRewardVideoAdVisible:(LYJAdRewardVideoAd *)rewardVideoAd {
    NSLog(@"激励视频展示");
}

- (void)lyjAdRewardVideoAdDidClick:(LYJAdRewardVideoAd *)rewardVideoAd {
    NSLog(@"激励视频点击");
}

- (void)lyjAdRewardVideoAdPlayFinished:(LYJAdRewardVideoAd *)rewardVideoAd {
    NSLog(@"激励视频播放完成");
}

- (void)lyjAdRewardVideoAdDidClose:(LYJAdRewardVideoAd *)rewardVideoAd {
```

```
    NSLog(@"激励视频关闭");
}
```

奖励建议以 `lyjAdRewardVideoAdRewarded:` 作为发奖依据，`lyjAdRewardVideoAdPlayFinished:` 仅表示播放完成。

10. 打开游戏能力

```
[LYJAdGameManager openGame:@"user_id"
                     extra:@{@"key": @"value"}
                completion:^(BOOL success, NSError * _Nullable error) {
    if (!success || error) {
        NSLog(@"打开游戏失败：%@", error);
    }
}];
```

指定游戏 ID：

```
[LYJAdGameManager openGame:@"user_id"
                    gameId:@"51100"
                     extra:@{@"key": @"value"}
                completion:^(BOOL success, NSError * _Nullable error) {
    if (!success || error) {
        NSLog(@"打开游戏失败：%@", error);
    }
}];
```

11. 常用调试信息

```
// 查看 SDK 版本
NSLog(@"%@", LYJAdSDKConfig.version);

// 查看当前成功链接的平台 SDK
NSLog(@"%@", [LYJAdSDKConfig.supportSDKs componentsJoinedByString:@","]);

// 查看广告平台名称
NSLog(@"sdkName = %@", ad.sdkName);
```

```
// 获取 eCPM，单位：分；-1 表示无权限或后台异常
NSInteger ecpm = [ad eCPM];

// 读取广告额外信息
NSDictionary *extraInfo = ad.extraInfo;
```

12. 接入注意事项

- 所有广告请求前必须先调用 `[LYJAdSDKConfig config:mediaVersion:]`。
- 广告对象需要被强引用保存，广告生命周期内不要提前释放。
- 展示类广告应在加载成功回调后调用 `showAd`。
- 模板信息流必须先 `render`，渲染成功后再展示 `adObject.view`。
- 信息流广告在可滚动容器中展示时，如后台配置百度广告，需要在广告进入可视区域后调用曝光上报方法。
- 原生广告复用或销毁前建议调用 `unregister` 解绑。
- 开屏广告底部自定义 View 可为空，若设置，高度建议不超过屏幕高度的 1/4。
- Demo 中广告位 ID 如 `1024`、`1025`、`117`、`501001`、`test-1229` 仅作示例，正式上线必须替换为正式广告位。
- 上线前建议关闭 `[LYJAdSDKConfig setDebug:YES]`。