## 畅游天际广告投放平台 SDK iOS

1. **Cocoapods集成**
2. 导入 CuskyAdSDK.xcframework 和 CuskyAdSDK.podspec 到项目下的 SDKs 文件夹下
3. pod 'CuskyAdSDK', :path => './SDKs/'
4. **手动集成**
5. 将 `CuskyAdSDK.framework` 拖入 Xcode 项目。
6. 在 Target 的 \*Build Phases\* 中确认已添加至 \*Link Binary With Libraries\*。
7. **配置**

为保证 SDK 的正常功能，需要在 `Info.plist` 中添加以下配置：

```xml

<key>NSUserTrackingUsageDescription</key>

<string>我们需要使用广告标识符以提供更精准的广告服务</string>

<key>NSLocationWhenInUseUsageDescription</key>

<string>我们需要使用您的位置来获取更准确的广告推荐</string>

<key>NSLocationAlwaysUsageDescription</key>

<string>我们需要使用您的位置来获取更准确的广告推荐</string>

```

1. **初始化**

无需额外初始化流程，直接调用 API 即可。

1. **加载广告**

基础请求

```objective-c

#import <CuskyAdSDK/CuskyAdSDK.h>

- (void)requestAd {

[CuskyAdSDKManager loadAdWithTagID:@"5"

caid:@""

layout:CuskyAdNativeLayoutFeed

videoInfo:nil

completion:^(AdResponseModel \*response, NSError \*error) {

if (!error) {

[self renderAd:response];

}

}];

}

```

视频广告请求（需配置 VideoInfo）

```objective-c

VideoInfo \*videoInfo = [[VideoInfo alloc] init];

videoInfo.width = 720; // 视频宽度

videoInfo.height = 1280; // 视频高度

videoInfo.type = 2; // 视频类型标识

videoInfo.mimes = @[@"video/mp4"]; // 支持格式

[CuskyAdSDKManager loadAdWithTagID:@"5"

caid:@""

layout:CuskyAdNativeLayoutRewardVideo

videoInfo:videoInfo

completion:^(AdResponseModel \*response, NSError \*error) {

if (!error) {

[self handleVideoAd:response.admobject.video];

}

}];

```

1. **广告展示**

原生广告渲染

```objective-c

- (void)renderAd:(AdResponseModel \*)adResponse {

// 确保在主线程执行UI操作

dispatch\_async(dispatch\_get\_main\_queue(), ^{

// 获取首个广告单元

Bid \*firstBid = adResponse.seatbid.bid.firstObject;

if (!firstBid) {

NSLog(@"⚠️ 无有效广告内容");

return;

}

// 清空旧内容

[self.adContainerView.subviews makeObjectsPerformSelector:@selector(removeFromSuperview)];

// 创建广告容器

UIView \*adView = [[UIView alloc] initWithFrame:self.adContainerView.bounds];

adView.backgroundColor = [UIColor whiteColor];

// 解析广告资源

[self parseNativeAssets:firstBid.admobject.native.assets inView:adView];

// 添加到展示容器

[self.adContainerView addSubview:adView];

});

}

#pragma mark - 原生图片加载（带加载指示器）

- (void)loadImageWithURL:(NSString \*)urlString imageView:(UIImageView \*)imageView {

NSURL \*url = [NSURL URLWithString:urlString];

if (!url) return;

// 添加加载指示器

UIActivityIndicatorView \*indicator = [[UIActivityIndicatorView alloc] initWithActivityIndicatorStyle:UIActivityIndicatorViewStyleGray];

[imageView addSubview:indicator];

[indicator startAnimating];

// 后台线程加载

dispatch\_async(dispatch\_get\_global\_queue(DISPATCH\_QUEUE\_PRIORITY\_DEFAULT, 0), ^{

NSData \*data = [NSData dataWithContentsOfURL:url];

UIImage \*image = [UIImage imageWithData:data];

// 主线程更新

dispatch\_async(dispatch\_get\_main\_queue(), ^{

imageView.image = image;

[indicator stopAnimating];

[indicator removeFromSuperview];

});

});

}

```

视频广告集成

```objective-c

- (void)setupVideoWithAsset:(NativeAsset \*)asset inView:(UIView \*)view {

if (!asset.video.url.length) return;

// 创建视频容器

UIView \*videoContainer = [[UIView alloc] initWithFrame:CGRectMake(0, 0, view.bounds.size.width, 200)];

// 初始化AVPlayer

AVPlayer \*player = [AVPlayer playerWithURL:[NSURL URLWithString:asset.video.url]];

AVPlayerLayer \*playerLayer = [AVPlayerLayer playerLayerWithPlayer:player];

playerLayer.frame = videoContainer.bounds;

playerLayer.videoGravity = AVLayerVideoGravityResizeAspect;

// 添加播放控制

AVPlayerViewController \*playerController = [[AVPlayerViewController alloc] init];

playerController.player = player;

playerController.showsPlaybackControls = YES;

// 嵌入视图

[videoContainer addSubview:playerController.view];

[view addSubview:videoContainer];

// 自动播放

[player play];

}

```

1. **曝光与点击**

上报接口说明

```objective-c

+ (void)reportClickWithAdResponse:(Bid \*)adResponse

view:(UIView \*)view

urlType:(AdUrlType)type

clickPoint:(CGPoint)clickPoint

completion:(void (^)(BOOL success, NSError \* \_Nullable error))completion;

```

- \*\*adResponse\*\*: 从 `AdResponseModel.seatbid.bid` 列表中获取单个 `Bid` 对象。

- \*\*view\*\*: 展示广告的 `UIView`。

- \*\*urlType\*\*: 上报类型，包括：

- `AdUrlTypeClick` （点击后上报点击的）

- `AdUrlTypeDeepLink` （跳转调用 deeplink\_urls）

- `AdUrlTypeImpression` （渲染成功调用上报曝光URL ）

- \*\*clickPoint\*\*: 相对于 `view` 左上角的点击坐标。

- \*\*completion\*\*: 上报完成回调。

示例：点击上报

```objective-c

- (void)handleAdTap:(UITapGestureRecognizer \*)tap {

CGPoint point = [tap locationInView:self.adContainerView];

Bid \*firstBid = [self.currentAdResponse.seatbid.firstObject.bid firstObject];

[CuskyAdSDKManager reportClickWithAdResponse:firstBid

view:self.adContainerView

urlType:AdUrlTypeClick

clickPoint:point

completion:^(BOOL success, NSError \*error) {

if (success) {

NSLog(@"点击上报成功");

} else {

NSLog(@"上报失败: %@", error.localizedDescription);

}

}];

}

```

1. **数据模型参考**

```objective-c

#pragma mark - 视频模型

@interface VideoContent : NSObject

@property (nonatomic, copy) NSString \*url;

@property (nonatomic, copy) NSString \*cover;

@property (nonatomic, assign) NSInteger duration;

@end

#pragma mark - 图片模型

@interface NativeImage : NSObject

@property (nonatomic, assign) NSInteger h;

@property (nonatomic, assign) NSInteger w;

@property (nonatomic, assign) NSInteger type;

@property (nonatomic, copy) NSString \*url;

@end

#pragma mark - 标题模型

@interface NativeTitle : NSObject

@property (nonatomic, copy) NSString \*text;

@end

#pragma mark - 数据模型

@interface NativeData : NSObject

@property (nonatomic, copy) NSString \*value;

@end

#pragma mark - 资源单元

@interface NativeAsset : NSObject

@property (nonatomic, assign) NSInteger assetId;

@property (nonatomic, assign) BOOL isrequired;

@property (nonatomic, strong) NativeImage \*img;

@property (nonatomic, strong) NativeTitle \*title;

@property (nonatomic, strong) VideoContent \*video;

@property (nonatomic, strong) NativeData \*data;

@end

#pragma mark - 原生广告内容

@interface NativeContent : NSObject

@property (nonatomic, strong) NSArray<NativeAsset \*> \*assets;

@end

#pragma mark - 广告对象

@interface AdmObject : NSObject

@property (nonatomic, strong) VideoContent \*video;

@property (nonatomic, strong) NativeContent \*native;

@end

#pragma mark - 事件跟踪

@interface TrackingEvents : NSObject

@property (nonatomic, strong) NSArray<NSString \*> \*imp\_urls;

@property (nonatomic, strong) NSArray<NSString \*> \*click\_urls;

@property (nonatomic, strong) NSArray<NSString \*> \*deeplink\_urls;

@end

#pragma mark - 竞价单元

@interface Bid : NSObject

@property (nonatomic, copy) NSString \*target;

@property (nonatomic, copy) NSString \*deeplink;

@property (nonatomic, copy) NSString \*action;

@property (nonatomic, copy) NSString \*bidId;

@property (nonatomic, copy) NSString \*crid; // 修改为NSString类型

@property (nonatomic, assign) NSInteger price;

@property (nonatomic, copy) NSString \*impid;

@property (nonatomic, strong) AdmObject \*admobject;

@property (nonatomic, copy) NSString \*adid;

@property (nonatomic, copy) NSString \*nurl;

@property (nonatomic, strong) TrackingEvents \*events;

@end

#pragma mark - 席位竞价

@interface SeatBid : NSObject

@property (nonatomic, strong) NSArray<Bid \*> \*bid;

@end

#pragma mark - 完整响应

@interface AdResponseModel : NSObject

@property (nonatomic, copy) NSString \*responseId;

@property (nonatomic, copy) NSString \*bidid;

@property (nonatomic, strong) SeatBid \*seatbid;

- (instancetype)initWithDictionary:(NSDictionary \*)dict;

@end

#pragma mark - 请求视频模型

@interface VideoInfo : NSObject

@property (nonatomic, assign) NSInteger width;

@property (nonatomic, assign) NSInteger height;

@property (nonatomic, assign) NSInteger type;

@property (nonatomic, strong) NSArray<NSString \*> \*mimes;

- (NSDictionary \*)toDictionary;

@end

```