

DU Ad Platform_SDK for iOS Access Guide

Version: DUAd_iOS_SDK_1.1.1

DU Ad Platform_SDK for iOS Access Guide

1. Introduction
2. Integration Workflow
3. Obtain Identity
 - 3.1 APP_ID
 - 3.2 DAP Placement ID
 - 3.3 Facebook Placement_ID(Optional)
 - 3.4 Admob Placement_ID(Optional)
4. Load SDK and Configuration
 - 4.1 Download the DU Ad Platform SDK Package
 - 4.2 Unzip the Package
 - 4.3 Load DU Ad Platform SDK when Using Xcode
5. Initialization
6. Maintain the status of user's collecting consent
 - 6.1 Set the status of user's consent
 - 6.2 Get the status of user's consent
7. Request Native Ad
 - 7.1 Declaration
 - 7.2 Set Delegate for Native Ad
 - 7.3 Retrieve Native Ad
8. Native Ad Properties
 - 8.1 Introduction of Ad Properties
 - 8.2 Get the Ad Properties
9. Register the Native Ad's View
10. Request Native list Ad
 - 10.1 Declaration
 - 10.2 Set Delegate for Native list Ad
 - 10.3 Retrieve Native list Ad
11. Request Interstitial Ad
 - 11.1 Declaration
 - 11.2 Set Delegate for Interstitial Ads
 - 11.3 Retrieve Interstitial Ad
12. Request Banner Ad
 - 12.1 Declaration
 - 12.2 Set Delegate for Banner Ads
 - 12.3 Retrieve Banner Ad

1. Introduction

This document describes how to integrate DU Ad Platform SDK into iOS apps. [DAP\(short for DUAd platform\)](#) offers advertising services for helping apps to monetize. This version of SDK provides native ads and interstitial ads.

Prerequisites:

DU Ad Platform SDK currently supports **iOS 8** (included) plus system versions. This SDK supports FacebookAudienceNetwork 4.99.0 (included) plus version.

2. Integration Workflow

This section describes the integration workflow of DU Ad Platform SDK.

1. Apply for App_ID and DAP Placement_ID.
2. Load SDK package.
3. Initialize DU Ad Platform SDK.
4. Access Du ads.

3. Obtain Identity

This section describes the three IDs needed during DU Ad Platform SDK integration: APP ID, DAP Placement ID, Facebook Placement_ID and Admob Placement_ID.

3.1 APP_ID

1. Definition

APP ID is a unique identifier of a developer's APP on **Du Ad Platform**. Each app has its own App ID.

2. Obtain method

Visit our [official website](#) and register your app on **Du Ad Platform**, the APP ID will be generated automatically

3. Code

```
@License
```

3.2 DAP Placement ID

1. Definition

DAP Placement ID is a unique identifier of an ad slot on Du Ad platform. Developers can create multiple DAP Placement IDs for one app.

2. Obtain method

Visit our [official website](#) and after registered your app, you can create the placement for your app.

3. Code

```
@pid
```

3.3 Facebook Placement_ID(Optional)

1. Definition

Facebook Placement ID is the unique identifier of an ad slot on Facebook audience network. Only required if you use DAP to intergrate FAN.

2. Obtain method

Visit [Facebook Developers](#) to apply it.

3. Code

```
@fbids
```

3.4 Admob Placement_ID(Optional)

1. Definition

Admob Placement ID is the unique identifier of an ad slot on Admob. Only required if you use DAP to intergrate Admob.

2. Obtain method

Visit [Admob Developers](#) to apply it.

3. Code

```
@amid
```

4. Load SDK and Configuration

This section describes how to load the DU Ad Platform SDK into your iOS project.

Please follow the below rules. Otherwise, there might be exceptions at run time.

4.1 Download the DU Ad Platform SDK Package

4.2 Unzip the Package

After unzipping ,the content is as below:

- File DUModuleSDK.framework:

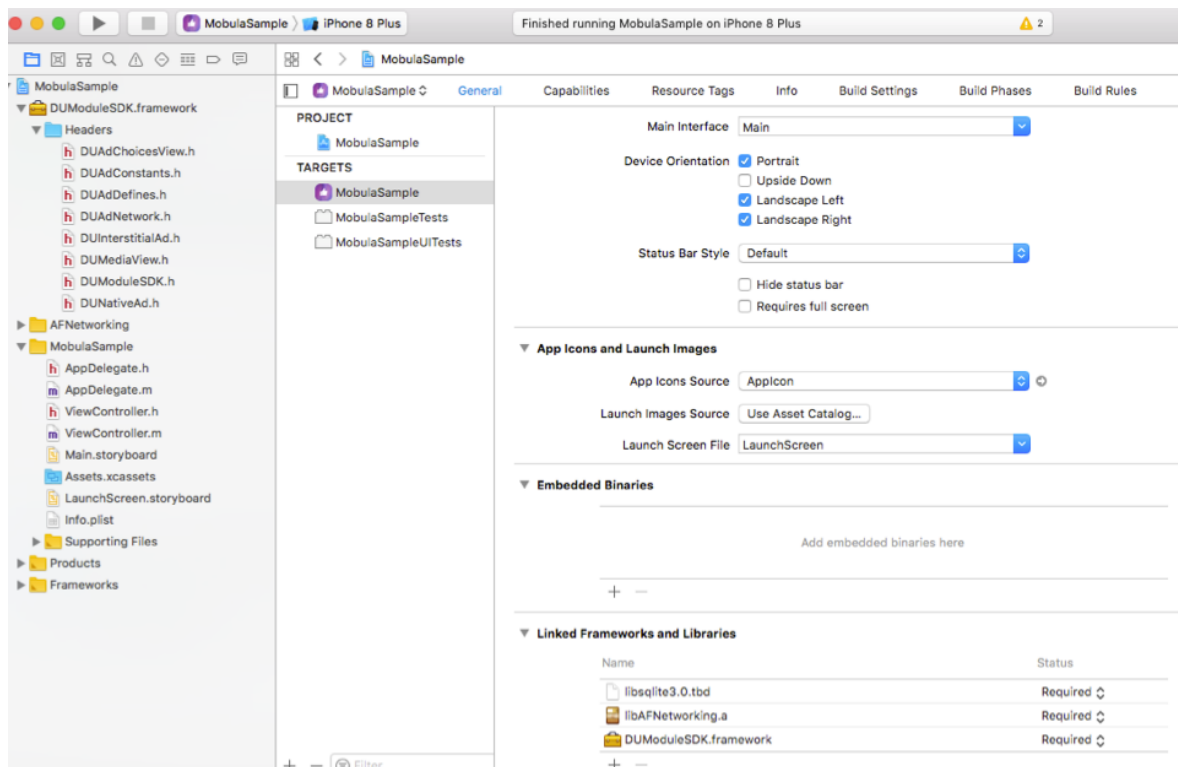
The DU Ad Platform SDK framework file

- MobulaSample

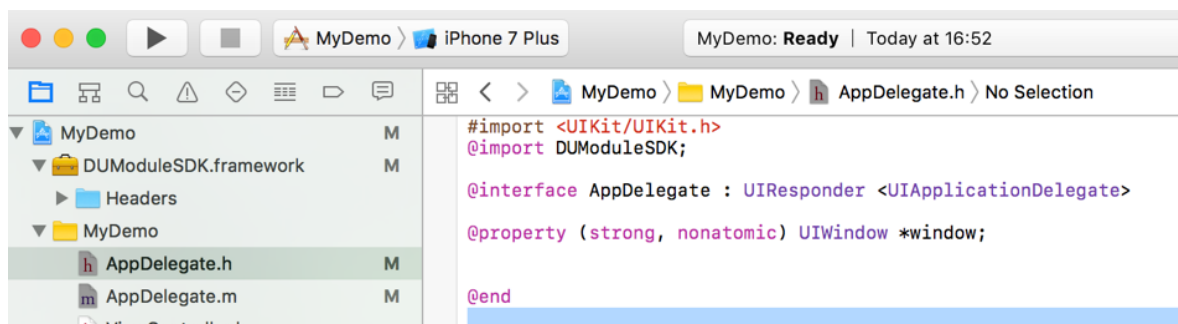
This folder stores a sample program, which integrates DU Ad Platform SDK. All interfaces in this document can be found in corresponding usage in this sample program.

4.3 Load DU Ad Platform SDK when Using Xcode

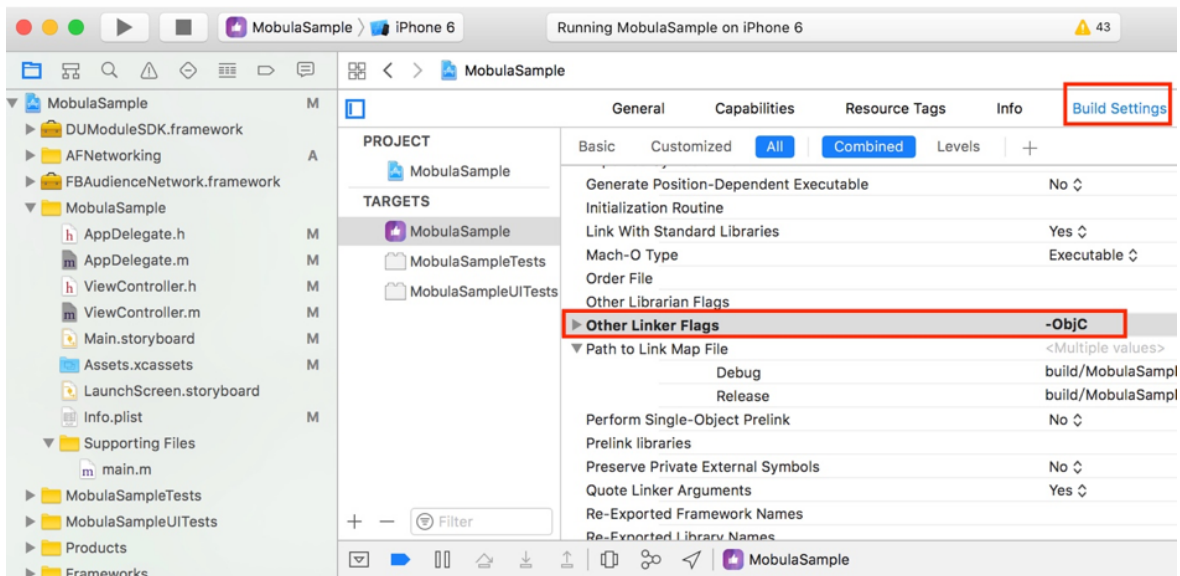
1. Add the libsqlite3.0.tbd into your project.
2. Drag the DUModuleSDK.framework to your app directory in Xcode. Select "Copy Items if needed" and click Finish. Then the DUModuleSDK.framework should be displayed in "Linked Framework and Libraries".



3. Import the SDK header DUModuleSDK in your AppDelegate.h file.



4. Go to the "Build Setting" of your project, Add "-ObjC" in "Other Linker Flags".



5. Initialization

This section describes how to initialize DAP SDK. You need to initialize DAP SDK before you can use it.

Placement id without initialization can not get ads.

1. Create a json file with mappings for the DAP Placement ID and other platform id as the code sample.

Note: If some of the DAP placements don't need ads from Facebook, the "fbids" part for that "pid" could be removed.

2. Add a call to `[DUAdNetwork initWithConfigDic: withLicense:]` in your AppDelegate.m file

Interface Instruction:

```
(void) initWithConfigDic: (NSDictionary*) aDic withLicense: (NSString*) aStr;
```

Parameters	Description
(NSDictionary*)aDic	The relationship between DAP Placement ID and other platform.
(NSString*)aStr	Your APP_ID of your app on Du Ad Platform.

3. Set the Log level to output more detailed information.

Note: Please remove method in release version.

Interface Instruction:

```
(void) setLogLevel: (DUAdLogLevel) aLevel;
```

Parameters	Description
DUAdLogLevelDebug	Debug mode to output full debug and error messages

Code Sample:

```

NSMutableDictionary *config=@{
    @"native" :
    @[
        @{
            @"pid" : @"YOUR_NATIVE_AD_PLACEMENT_ID" ,
            @"fbids" : @[@"YOUR_FBID"] ,
        },
        @{
            @"pid" : @"YOUR_INTERSTITIAL_AD_PLACEMENT_ID" ,
            @"fbids" : @[@"YOUR_FBID"] ,
        }
    ]
};
[DUAdNetwork initWithConfigDic:config
withLicense:@"YOUR_DAP_APP_LICENSE"];
[DUAdNetwork setLogLevel:DUAdLogLevelDebug];

```

6. Maitain the status of user's collecting consent

This is an optional configuration for GDPR compliance.

6.1 Set the status of user's consent

Please set the status during the initialization.

Interface Instrunction:

```
(void)setConsentStatus:(BOOL)userConsent;
```

Parameters	Description
True	Obtained user's consent, the advertising service can be used as usual
False	No user's consent, all advertising services will be stopped

6.2 Get the status of user's consent

Interface Instrunction:

```
(BOOL)getConsentStatus;
```

Obtain the status of user's consent. Return True if user's consent is obtained, otherwise return False.

7. Request Native Ad

7.1 Declaration

Import the DUModuleSDK in your View Controller header file, declare that ViewController implements the DUNativeAdDelegate protocol and add DUNativeAd instance variable:

Proceed as follows:

1. Create DuNative Ad Object.

Must specify the corresponding pid into the object. You will get different ad data with different pid.

2. Set ad cache size

Cache size could be set to 1-5. Recommend not to set cachesize. The default cachesize will be 1.

Note:Cache size only takes effect when integrating other platform through the DU Ad Platform.

Interface Instructions:

```
(nonnullinstancetype)initWithPlacementID:(nonnull NSString*)placementID;
```

```
(nonnullinstancetype)initWithPlacementID:(nonnull NSString*)placementID cacheSize:  
(NSInteger)aSize;
```

Parameters	Description
(nonnull NSString*)placementID	DAP placement ID, this pid must declared on json's native array
(NSInteger)aSize	Ad cache size.

Code Sample:

```

#import <UIKit/UIKit.h>
#import DUModuleSDK; /*DU SDK*/
#import <FBAudienceNetwork/FBAudienceNetwork.h> /*FacebookSDK*/
@interface ViewController : UIViewController <DUNativeAdDelegate>
@property (strong, nonatomic) DUNativeAd *nativeAd;
@end

-(void)viewDidLoad
{
    [super viewDidLoad];
    DUNativeAd _nativeAd = [[DUNativeAd alloc] initWithPlacementID: @"88888"
cacheSize:1];
    _nativeAd.delegate = self;
}

```

7.2 Set Delegate for Native Ad

Please register a callback interface for receiving the native ad data.

Interface Instruction:

```
@protocol DUNativeAdDelegate <NSObject>
```

```

@protocol DUNativeAdDelegate <NSObject>
@optional

/*Retrieve ad successfully after calling loadAd*/
- (void)nativeAdDidLoad:(nonnull DUNativeAd *)nativeAd;

/*Sent immediately before the impression of an NativeAd object will be
logged*/
- (void)nativeAdWillLogImpression:(nonnull DUNativeAd *)nativeAd;

/*Get an error*/
- (void)nativeAd:(nonnull DUNativeAd *)nativeAd didFailWithError:(nonnull
NSError *)error;

/*Retrieve a ad click event*/
- (void)nativeAdDidClick:(nonnull DUNativeAd *)nativeAd;

/*When an ad is clicked, the modal view will be presented. And when the
user finishes the interaction with the modal view and dismiss it, this
message will be sent, returning control to the application.*/
- (void)nativeAdDidFinishHandlingClick:(nonnull DUNativeAd *)nativeAd;

@end

```


Error code and description are shown as below:

Constants	Error Code	Description
NETWORK_ERROR_CODE	1000	Client network error
NO_FILL_ERROR_CODE	1001	No Ad data retrieved
LOAD_TOO_FREQUENTLY_ERROR_CODE	1002	Too many interface requests
IMPRESSION_LIMIT_ERROR_CODE	1003	Reach the daily impression limit
SERVER_ERROR_CODE	2000	Server error
INTERNAL_ERROR_CODE	2001	Network error
MISSING_PROPERTIES_CODE	2002	Missing Properties
TIME_OUT_CODE	3000	Retrieve Ad data timed out
UNKNOWN_ERROR_CODE	3001	Unknown error
NO_CHANNEL_ERROR_CODE	3002	No available channel
NO_USER_CONSENT_ERROR_CODE	4000	No user's consent

7.3 Retrieve Native Ad

The ad retrieving process is asynchronous, so it will not block developers' threads.

Interface Instructions:

(void) fillAd

Use the `fillAd` to pre-cache ad in advance for faster loading the ad when using `loadAd`. Recommend using the `fillAd` at the page before the ad showing page.

Note: Ad data will be cached in client device's memory. Since SDK only caches the image's URL address not the image data, the cache size is small.

(void) loadAd

Acquire advertising object data **asynchronously**, making an ad request when there is no cache.

Recommend using `fillAd` after `loadAd` to pre-cache again.

(DUNativeAd*) getCacheAd

Acquire advertising object data **synchronously**. It could be traversed until the number of cached ads goes to 0.

Please make sure the cache pool is not null before showing ad.

Recommend using `fillAd` after `getCacheAd` to pre-cache again.

(BOOL) isHasCached

Check if there is cached ad. Return true for having cache.

Code Sample:

```
@interface ViewController ()<DUNativeAdDelegate>
{
    DUNativeAd *_nativeAd;
    DUMediaView *_mediaView;
    DUAdChoicesView *_chioceView;
}

- (IBAction)loadAd:(id)sender
{
    NSLog(@"load Native ad.");
    [_nativeAd loadAd];
}

- (void)nativeAdDidLoad:(DUNativeAd *)nativeAd
{
    NSLog(@"Native ad was loaded.");
    [self nativeAdDisplay:nativeAd];
}

- (void)nativeAdWillLogImpression:(DUNativeAd *)nativeAd
{
    NSLog(@"Native ad impression is being captured.");
}

- (void)nativeAd:(DUNativeAd *)nativeAd didFailWithError:(NSError *)error
{
    NSLog(@"Native ad failed to load with error: %@", error);
}

- (void)nativeAdDidClick:(DUNativeAd *)nativeAd
{
    NSLog(@"Native ad was clicked.");
}

- (void)nativeAdDidFinishHandlingClick:(DUNativeAd *)nativeAd
```

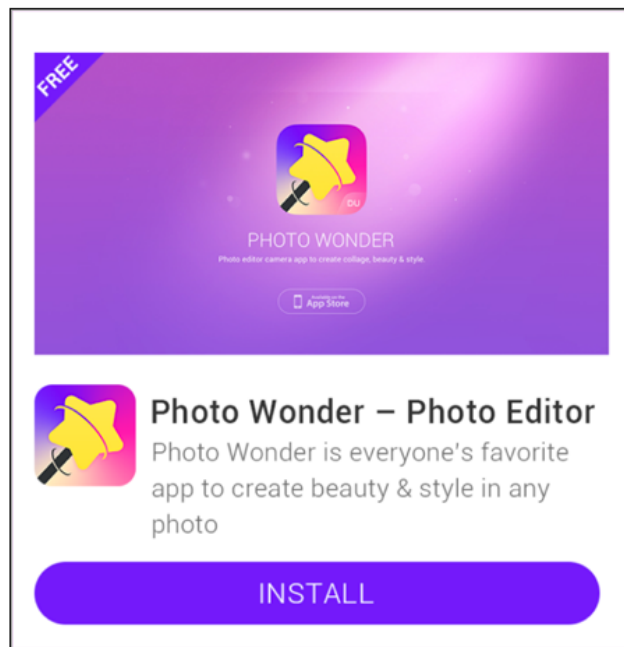
```
{
    NSLog(@"Native ad did finish click handling.");
}
```

8. Native Ad Properties

When using the Native Ad, instead of receiving an ad ready to be displayed, you will receive a group of ad properties such as a title, an image, a call to action, and you will have to use them to construct a custom view where the ad is shown. This section describes the ad properties and how to get them.

8.1 Introduction of Ad Properties

Ad properties usually include: Icon, title, Call to action (CTA) button, short description, promotion image, etc.



8.2 Get the Ad Properties

The interfaces for retrieving the ad properties as shown below:

- Get Icon

```
@property (nonatomic, strong, readonly, nullable) NSString *iconUrl
```

Return the URL address of icon.

- Get Title

```
@property (nonatomic, copy, readonly, nullable) NSString *title
```

Return the title of ad.

Please reserve at least 20 characters' space to display the title. An ellipsis (...) can be used to indicate truncated text.

Note: The ad title must be included in your native ad design.

- Get Call to Action (CTA) button

```
@property (nonatomic, copy, readonly, nullable) NSString *callToAction
```

Return the text of ad's CTA button.

Advertisers can specify the text of CTA button, e.g. Install Now. The max character length is 25. Please do not shorten or change the text.

Note: The CTA button must be included in your native ad design.

- Get Short Description

```
@property (nonatomic, copy, readonly, nullable) NSString *shortDesc
```

Return the short description of ad.

Please reserve at least 72 characters' space to display the short description. If the space is not big enough, it is recommended to use scrolling text effects, or do not display the short description.

- Get Promotion Image

```
@property (nonatomic, strong, readonly, nullable) NSString *imageUrl
```

Return the URL address of ad's promotion image. NULL for no image.

A promotion image can increase user's desire to click the ad. The image size is usually 796x416 pixels(1.91:1). You can zoom and cut part of the image, but do not distort or change it.

- DuAdChoicesView

This view is the AdChoices corner mark from by Facebook Native Ad. It's the mandatory element for Facebook native Ad. Please Note that the native ad which is not from Facebook doesn't have this.

Code Sample:

```
DUAdChoicesView *choiceView = [[DUAdChoicesView alloc]
initWithNativeAd:nativeAd expandable:NO];
[self.adChoicesView addSubview:_chioceView];
```

- Ad channel type

Indicate the ad source channel.

```
@property (nonatomic, assign, readonly) DUAdChannelType adChannelType
```

Value	Description
DUAdChannelTypeUnknow	Ads from unknown source
DUAdChannelTypeDownload	Ads from DAP
DUAdChannelTypeFacebook	Ads from Facebook Audience Network

9. Register the Native Ad's View

The SDK will log the impression and handle the click automatically. Please note that you must register the ad's view with the nativeAd instance to enable that.

Interface Instructions:

```
(void)registerViewForInteraction:(UIView *)view mediaView:(nonnull id)mediaView
iconView:(nullable id)imageView viewController:(nullable UIViewController
*)viewController;
```

```
(void)registerViewForInteraction:(UIView *)view mediaView:(nonnull id)mediaView
iconView:(nullable id)imageView viewController:(nullable UIViewController
*)viewController clickableViews:(nullable NSArray<UIView *> *)clickableViews;
```

Parameters	Description
(UIView *)view mediaView	Media view
(nullable id)imageView	Icon view, Facebook icon view is available
(UIViewController *)viewController	More detailed sub-View
(NSArray<UIView *> *)clickableViews	Clickable View in Ad contents

Unregister the view:

```
(void)unregisterView
```

10. Request Native list Ad

Du Native Ad List is for showing multiple ads in one page at same time.

10.1 Declaration

Import the DUModuleSDK in your View Controller header file, declare that ViewController implements the DUNativeAdsManagerDelegate protocol and add DUNativeAdsManager instance variable:

Steps:

1. Create native list Ad Object.

Must specify the corresponding pid into the object. You will get different ad data with different pid.

2. Set ad cache size

Cache size could be set to 1-10. The default cachesize will be 10.

Interface Instructions:

```
(instancetype)initWithPlacementID:(NSString*)placementID;  
(instancetype)initWithPlacementID:(NSString *)placementID  
cacheSize:(NSInteger)aSize;
```

Parameters	Description
(NSString*)placementID	DAP placement ID, it must be declared in Json's list array, see chapter 5.
(NSInteger)aSize	Ad cache size. default value is 10.

Code Sample:

```
#import <UIKit/UIKit.h>  
@import DUModuleSDK;  
  
@interface ViewController : UIViewController <DUNativeAdsManagerDelegate,  
DUNativeAdDelegate>  
    @property (strong, nonatomic) DUNativeAdsManager *adsMgr;  
@end  
  
- (void) viewDidLoad {  
    [super viewDidLoad];  
    /*!  
    @method  
    @abstract
```

```
This is a method to initialize a DUNativeAdsManager object matching the
given placement id. This will use 10 as default cache size.
@param placementID The id of the ad placement. You can create your
placement id from Mobula developers page.
*/

_adsMgr = [[DUNativeAdsManager alloc] initWithPlacementID:@"10035"];
_adsMgr.delegate = self;
}
```

10.2 Set Delegate for Native list Ad

Please register a callback interface for receiving the native list ad data.

Interface Instruction:

```
@protocol DUNativeAdsManagerDelegate
```

```
@protocol DUNativeAdsManagerDelegate <NSObject>
@optional

/*Retrieve ad successfully after calling loadAd*/
- (void)nativeAdsLoaded:(NSArray<DUNativeAd *> *)nativeAds;

/*Get an error*/
- (void)nativeAdsFailedToLoadWithError:(NSError *)error;

@end
```

Error code and description are shown as below:

Constants	Error Code	Description
NETWORK_ERROR_CODE	1000	Client network error
NO_FILL_ERROR_CODE	1001	No Ad data retrieved
LOAD_TOO_FREQUENTLY_ERROR_CODE	1002	Too many interface requests
IMPRESSION_LIMIT_ERROR_CODE	1003	Reach the daily impression limit
SERVER_ERROR_CODE	2000	Server error
INTERNAL_ERROR_CODE	2001	Network error
MISSING_PROPERTIES_CODE	2002	Missing Properties
TIME_OUT_CODE	3000	Retrieve Ad data timed out
UNKNOWN_ERROR_CODE	3001	Unknown error
NO_CHANNEL_ERROR_CODE	3002	No available channel
NO_USER_CONSENT_ERROR_CODE	4000	No user's consent

10.3 Retrieve Native list Ad

The ad retrieving process is asynchronous, so it will not block developers' threads.

Interface Instructions:

(void) fillAds

Use the `fillAds` to pre-cache ad in advance for faster loading the ad when using `loadAds`. Recommend using the `fillAds` at the page before the ad showing page.

Note: Ad data will be cached in client device's memory. Since SDK only caches the image's URL address not the image data, the cache size is small.

(void) loadAds

Acquire advertising object data **asynchronously**, making an ad request when there is no cache.

Recommend using `fillAds` after `loadAds` to pre-cache again.

(NSArray<DUNativeAd *> *)getCacheAds

Acquire advertising object data **synchronously**. It could be traversed until the number of cached ads goes to 0.

Please make sure the cache pool is not null before showing ad.

Recommend using `fillAds` after `getCacheAds` to pre-cache again.

(BOOL) isHasCached

Check if there is cached ad. Return true for having cache.

Code Sample:

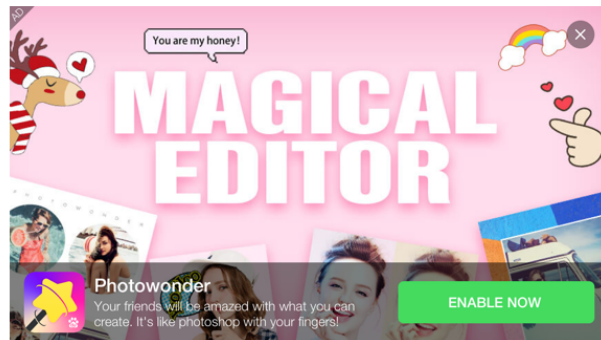
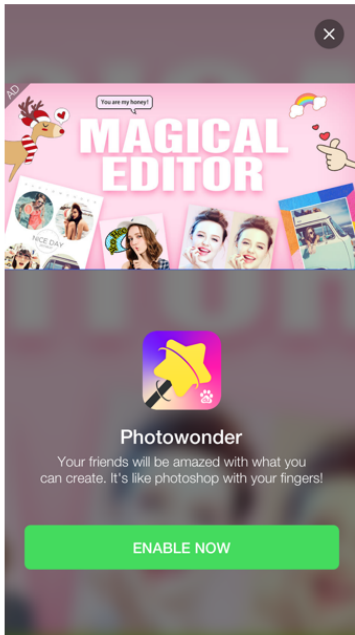
```
@interface NativeListAdViewController () < DUNativeAdsManagerDelegate,
DUNativeAdDelegate> {
    DUNativeAdsManager *adsMgr;
    NSMutableArray<DUNativeAd *> *m_adArray;
}

-(IBAction)load:(id)sender {
    NSLog(@"Native list ads were loaded. ");
    [adsMgr loadAds];
}

- (void)nativeAdsFailedToLoadWithError:(nonnull NSError *)error {
    NSLog(@"Native list ads failed to load with error: %@", error);
}

- (void)nativeAdsLoaded:(nonnull NSArray<DUNativeAd *> *)nativeAds {
    NSInteger adCount = 0;
    if (nativeAds) {
        adCount = [nativeAds count];
    }
    if (adCount > 0) {
        [m_adArray addObjectsFromArray:nativeAds];
        [self showAds];
    }
    NSLog(@"nativeAdsLoaded adCount : %ld, and cache : %@", adCount,
nativeAds);
}
```

11. Request Interstitial Ad



11.1 Declaration

Import the DUModuleSDK in your View Controller header file, declare that ViewController implements the DUInterstitialAdDelegate protocol and add DUInterstitialAd instance variable:

Proceed as follows :

1. Create DUInterstitialAd Object.

Must specify the corresponding pid into the object. You will get different ad data with different pid.

2. Set ad cache size

Cache size could be set to 1-5. Recommend not to set cachesize. The default cachesize will be 1.

Note: Cache size only takes effect when integrating other platform through the DU Ad Platform.

Interface Instructions:

```
(nonnull instancetype)initWithPlacementID:(nonnull NSString*)placementID;
(nonnull instancetype)initWithPlacementID:(nonnull NSString*)placementID cacheSize:
(NSInteger)aSize;
```

Parameters	Description
(nonnull NSString*)placementID	DAP placement ID, this pid must declared on json's native array
(NSInteger)aSize	Ad cache size.

If you need DAP to mediate Admob' interstitial ad, please follow the below format to configure the json.

```

{
  @"amaid" : "YOUR_ADMOB_APP_ID",
  @"native" :
    @[
      @{
        @"pid" : @"YOUR_DAP_INTERSTITIAL_PLACEMENT_ID",
        @"fbids" : @[@"YOUR_FACEBOOK_INTERSTITIAL_PLACEMENT_ID"],
        @"amid" : @"YOUR_ADMOB_INTERSTITIAL_PLACEMENT_ID",
      }
    ]
}

```

Code Sample:

```

#import <UIKit/UIKit.h>
#import DUModuleSDK; /*DU SDK*/
#import <FBAudienceNetwork/FBAudienceNetwork.h> /*FacebookSDK*/
@interface ViewController : UIViewController <DUInterstitialAdDelegate>
@property (strong, nonatomic) DUInterstitialAd *interstitialAd;
@end

-(void)viewDidLoad
{
    [super viewDidLoad];
    DUInterstitialAd _interstitialAd= [[DUInterstitialAd alloc]
initWithPlacementID: @"88888" cacheSize:1];
    _interstitialAd.delegate = self;
}

```

11.2 Set Delegate for Interstitial Ads

Please register a callback interface for receiving the ad data.

Interface Instructions:

```
@protocol DUInterstitialAdDelegate <NSObject>
```

```

@optional

/*Retrieve ad successfully after calling loadAd*/
- (void)interstitialAdDidLoad:(DUInterstitialAd *)interstitialAd;

```

```

/*Sent immediately before the impression of an NativeAd object will be
logged*/
- (void)interstitialAdWillLogImpression:(DUInterstitialAd *)interstitialAd;

/*Get an error*/
- (void)interstitialAd:(DUInterstitialAd *)interstitialAd didFailWithError:
(NSError *)error;

/*Retrieve a ad click event*/
- (void)interstitialAdDidClick:(DUInterstitialAd *)interstitialAd;

/*Sent immediately before the Ad object will be closed*/
- (void)interstitialAdWillClose:(DUInterstitialAd *)interstitialAd;

/*Sent after the Ad object closed*/
- (void)interstitialAdDidClose:(DUInterstitialAd *)interstitialAd;
@end

```

11.3 Retrieve Interstitial Ad

The ad retrieving process is asynchronous, so it will not block developers' threads.

Interface Instructions:

(void) fillAd

Use the `fillAd` to pre-cache ad in advance for faster loading the ad when using `loadAd`. Recommend using the `fillAd` at the page before the ad showing page.

Note: Ad data will be cached in client device's memory. Since SDK only caches the image's URL address not the image data, the cache size is small.

(void) loadAd

Acquire advertising object data **asynchronously**, making an ad request when there is no cache.

Recommend using `fillAd` after `loadAd` to pre-cache again.

(DUInterstitialAd*) getCacheAd

Acquire advertising object data **synchronously**. It could be traversed until the number of cached ads goes to 0.

Please make sure the cache pool is not null before showing ad.

Recommend using `fillAd` after `getCacheAd` to pre-cache again.

(BOOL) isHasCached

Check if there is cached ad. Return true for having cache.

(BOOL)showAdFromRootViewController:(nullable UIViewController *) rootViewController

Show interstitial ads.

12. Request Banner Ad

A sample of banner ad



12.1 Declaration

Import the DUModuleSDK in your View Controller header file, declare that ViewController implements the DUBannerAdViewDelegate protocol and add DUBannerAdView instance variable.

Note: Banner Ad currently don't support intergrating other platform.

Proceed as follows :

1. Create DUBannerAdView Object.

Please create the object on the method `viewDidLoad` in your View Controller.

2. Set the location of Banner View

Interface Instructions:

(instancetype)initWithPlacementID:(NSString *)placementID adSize:

(DUBannerAdSize)adSize rootViewController:(nullable UIViewController *)viewController;

Parameters	Description
(NSString *)placementID	DAP placement ID, this pid must declared on Json's native array
(DUBannerAdSize)adSize	Set Ad size. Supported Value : <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">kDUBannerAdSize320x50</div> : size with 320 x 50 px
(nullable UIViewController *)viewController	The view controller that will be used to present the ad

Code Sample:

```
#import <UIKit/UIKit.h>
#import DUModuleSDK; /*DU SDK*/

@interface ViewController : UIViewController <DUBannerAdViewDelegate>
    @property (strong, nonatomic) DUBannerAdView *bannerAd;
@end

- (void)viewDidLoad {
    [super viewDidLoad];
    _bannerAd=[[DUBannerAdView alloc] initWithPlacementID:PID_BANNER
adSize: kDUBannerAdSize320x50 rootViewController:self];
    _bannerAd.frame = CGRectMake(x-coordinate, y-coordinate,
adView.bounds.size.width, adView.bounds.size.height);
    _bannerAd.delegate=self;
    [self.view addSubview:_bannerAd];
}
```

12.2 Set Delegate for Banner Ads

Please register a callback interface for receiving the ad data.

Interface Instruction:

```
@protocol DUBannerAdViewDelegate <NSObject>
```

```
@protocol DUBannerAdViewDelegate <NSObject>
@optional

/*Retrieve ad successfully after calling loadAd*/
- (void)adViewDidLoad:(DUBannerAdView *)bannerAdView;

/*Get an error*/
- (void)adView:(DUBannerAdView *)bannerAdView didFailWithError:(NSError
*)error;
@end
```

12.3 Retrieve Banner Ad

The ad retrieving process is asynchronous, so it will not block developers' threads.

Interface Instruction:

```
(void)loadAd;
```