

DU Ad Platform SDK for IOS Access Guide

DUAd_iOS_SDK_1.0.5

Baidu Online Network Technology (Beijing) Co., Ltd

No.	DUAd10120150810
Date	2017-11-21
Ver.	1.0.5
Email	support_duad@baidu.com

Contents

1	Introduction.....	1
1.1	Target Audience.....	1
1.2	Prerequisites	1
2	Integration Workflow	1
3	Obtain Identity.....	2
3.1	APP ID	2
3.2	DAP Placement ID	2
3.3	Facebook Placement ID	2
4	Load SDK and Configuration	3
4.1	Load DU Ad Platform SDK.....	3
5	Initialization.....	4
6	Request native ad	6
6.1	Declaration.....	6
6.2	Construction.....	6
6.3	Pre-cache the native ad.....	7
6.4	Retrieve native ad.....	7
6.4.1	Set delegate for native ad.....	7
6.4.2	Retrieve ad	7
6.5	Get cached native ad.....	9
6.5.1	Check if there is cached ad	9
6.5.2	Get cached native ad	9
7.	Native ad properties	9
7.1	Introduction of ad properties	9
7.2	Ad properties.....	10
8.	Register the native ad's View	12
9.	Register interstitial ad	13
9.1	Declaration.....	13
9.2	Construction.....	13
9.3	Pre-cache the interstitial ad.....	14
9.4	Retrieve interstitial ad.....	14
9.4.1	Set delegate for interstitial ad.....	14
9.4.2	Retrieve ad	15
9.4.3	Show ad.....	15
9.5	Get cached native ad.....	15
9.5.1	Check if there is cached ad	15
9.5.2	Get cached native ad	15

1 Introduction

This document describes how to integrate **DU Ad Platform SDK** into IOS apps. **DAP, short for DU Ad platform** offers advertising services for helping IOS apps to monetize. This version of SDK provides native ads.

1.1 Target Audience

This document is for IOS app developers.

1.2 Prerequisites

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

2 Integration Workflow

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

- The integration workflow for **Du Native Ad**:
 1. Apply for App_ID and DAP Placement_ID and Facebook Placement ID. See [Section 3](#).
 2. Load **DU Ad Platform SDK** package. See [Section 4](#).
 3. Initialize **DU Ad Platform SDK** See [Section 5](#).
 4. Access Du native ad. See [Section 6](#). [Section 7](#). [Section 8](#).

- The integration workflow for **Du Interstitial Ad**:
 1. Apply for App_ID and DAP Placement_ID and Facebook Placement ID. See [Section 3](#).
 2. Load **DU Ad Platform SDK** package. See [Section 4](#).
 3. Initialize **DU Ad Platform SDK** See [Section 5](#).
 4. Access Du interstitial ad. See [Section 9](#).

3 Obtain Identity

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

3.1 APP ID

A. Definition

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

B. Obtain method

Visit our official website <http://ad.duapps.com> and register your app on **Du Ad Platform**, the APP ID will be generated automatically.

C. Code

License

3.2 DAP Placement ID

A. Definition

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

B. Obtain method

Visit our official website <http://ad.duapps.com> and after registered your app, you can create the placement for your app.

C. Code

pid

3.3 Facebook Placement ID

A. Definition

Facebook Placement ID is the unique identifier of an ad slot on Facebook audience network.

B. Obtain method

Visit Facebook Developers <https://developers.facebook.com> to apply it.

C. Code

fbids

4 Load SDK and Configuration

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

4.1 Load DU Ad Platform SDK

A. **Download** the DU Ad Platform SDK package.

- Package name: [DUAd_iOS_SDK_1.0.5.zip](#)

B. **Unzip** the package

After unzipping the package, two folders are available in the subdirectory:

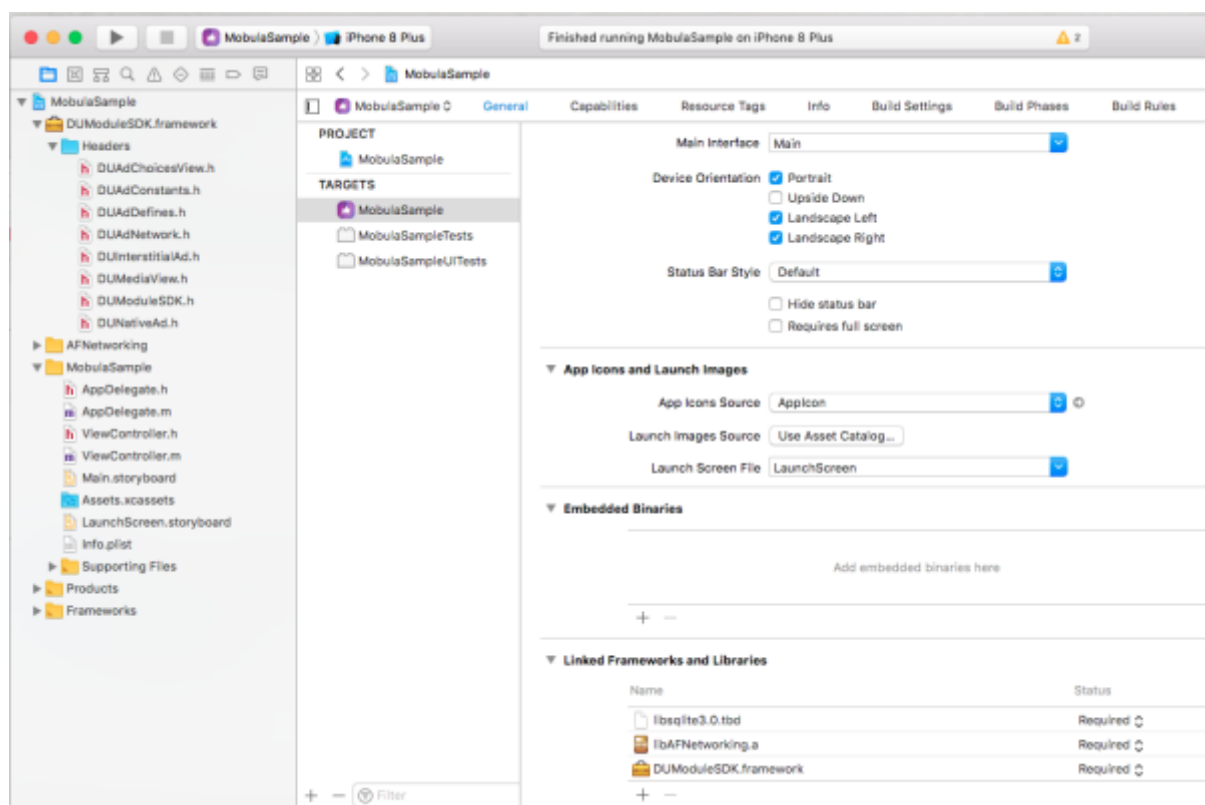
- **DUAd_SDK**
[DUModuleSDK.framework](#)

- **DUAd_SDK_DEMO**

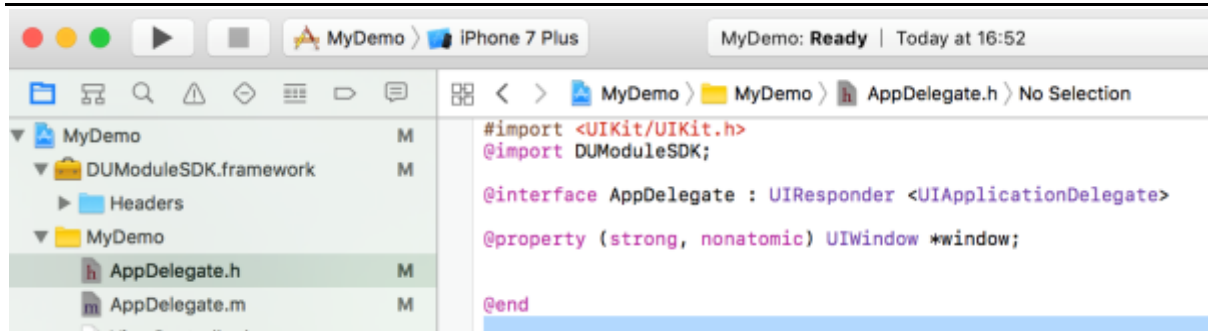
This folder stores a sample program, which integrates **DUAd SDK**. All interfaces in this document can be found in corresponding usage in this sample program.

C. **Load** DU Ad Platform SDK when using Xcode

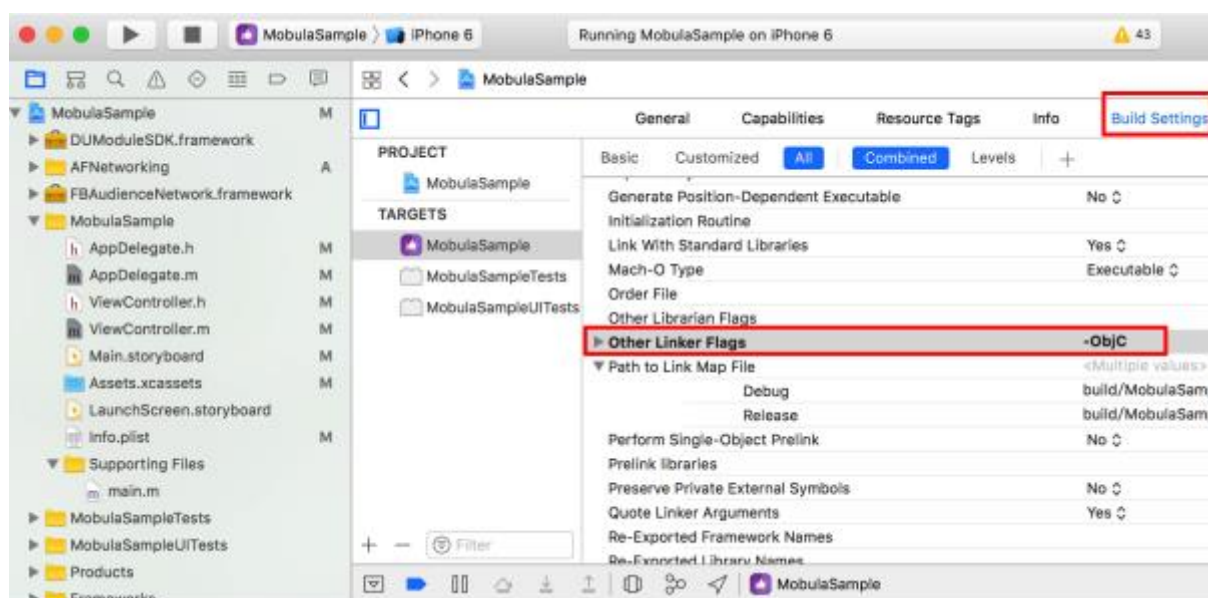
- 1) Add the [libsqlite3.0.tbd](#)
- 2) Drag the [DUModuleSDK.framework](#) to your app directory in Xcode. Select "Copy Items if needed" and click Finish. Then the [DUModuleSDK.framework](#) should be displayed in "Linked Frameworks and Libraries"



3) Now, in your AppDelegate.h file, import the SDK header [DUModuleSDK](#).



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.

- **Method:**

In your AppDelegate.m file, call the following interface to finish the initialization.
 [DUAdNetwork initWithConfigDic: withLicense:];

- **Interface Instruction:**

```

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

Parameters	Description
(NSDictionary*)aDic	The relationship between DAP Placement ID and Facebook Placement ID.
(NSString*)aStr	It is your APP ID on DAP, a unique identifier of your APP on Du Ad Platform . See 3.1

- **JSON format:**

Using JSON format to write **String** data with mappings for the **DAP Placement ID (pid)** and **Facebook Placement ID (fbids)**

```
{
  "native": [
    {
      "pid": "YOUR_DAP_PLACEMENT_ID",
      "fbids": [
        "YOUR_FACEBOOK_PLACEMENT_ID"
      ]
    },
    {
      "pid": "YOUR_DAP_PLACEMENT_ID"
    }
  ]
}
```

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

- **Code Sample:**

```
NSMutableDictionary *config=@{
    @"native" :
        @[
            @{
                @"pid" : @"888888" ,
                @"fbids" : @[@"888888_888888"] ,
            },
            @{
                @"pid" : @"666666",
                @"fbids" : @[@"666666_666666"] ,
            }
        ]
};

[DUAdNetwork initWithConfigDic: config withLicense:@"888dd888888d "];
```

6 Request native ad

6.1 Declaration

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

- **Code Sample:**

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

6.2 Construction

- **Interface Instructions:**

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

Parameters	Description
int pid	DAP placement ID, see 3.2
int cacheSize	ad cache size. Cachesize could be set to 1-5. Recommend not to set cachesize. The default cachesize will be 1.

- **Code Sample:**

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


6.3 Pre-cache the native ad

- **Interface Instruction:**

- (void)fillAd;

Use the fillAd to pre-cache ads in advance, for faster loading the ads when using loadAd.

Suggestion: Use 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.

6.4 Retrieve native ad

Please register a callback interface for receiving the native ad data. The ad retrieving process is asynchronous, so it will not block developers' threads.

6.4.1 Set delegate for native ad

- **Interface Instruction:**

@protocol DUNativeAdDelegate <NSObject>

Description
<pre> /* 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; </pre>

6.4.2 Retrieve ad

- **Interface Instruction:**

- (void)loadAd;

- **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.");
}

```

Constants	Table Error Code	
	Error Code	Description
<i>NETWORK_ERROR_CODE</i>	1000	Network error

<i>NO_FILL_ERROR_CODE</i>	1001	No Ad data retrieved
<i>LOAD_TOO_FREQUENTLY_ERROR_CODE</i>	1002	Too many interface requests
<i>SERVER_ERROR_CODE</i>	2000	Server error
<i>MISSING_PROPERTIES_CODE</i>	2002	Missing Properties
<i>TIME_OUT_CODE</i>	3000	Retrieve Ad data timed out
<i>UNKNOW_ERROR_CODE</i>	3001	Unknown error
<i>NO_CHANNEL_ERROR_CODE</i>	3002	No available channel

6.5 Get cached native ad

6.5.1 Check if there is cached ad

- **Interface Instruction:**
- (BOOL)isHasCached;

6.5.2 Get cached native ad

- **Interface Instruction:**
- (DUNativeAd*)getCacheAd;

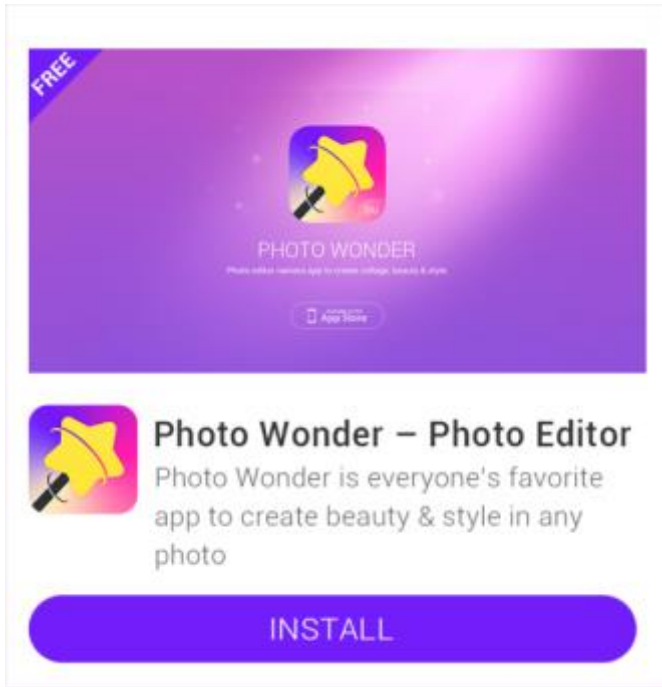
This interface is for getting the cached ad for the current ad object. It could be traversed until the number of cached ads goes to 0. Please make sure the cache pool is not null before showing ad.

7. 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.

7.1 Introduction of ad properties

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



7.2 Ad properties

The ad properties as shown below:

- **Icon**

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

Description
The URL address of icon

- **Title**

Please reserve at least 20 characters' space to display the title.

An ellipsis (...) can be used to indicate truncated text.

Please note the ad title must be included in your native ad design.

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

Description
The title of ad

- **Call to Action (CTA) button**

Advertisers can specify the text of CTA button, e.g. **Install Now**. Please do not shorten or change the text.

For CTA button with promotion image, the **max** character length is **25**. For CTA button without image, the text is usually defined as **Download**.

Please note the CTA button must be included in your native ad design.

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

Description
The text of ad's CTA button

- **Short description**

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.

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

Description
The short description of ad

- **Promotion Image**

A promotion image can increase user's desire to click the ad.

The image size is usually: 1200x628 pixels. You can zoom and cut part of the image, but do not distort or change it. **Please note** that not all ads have promotion images.

@property (nonatomic, strong, readonly, nullable) NSString *imgUrl;

Description
The URL address of ad's promotion image.
When the image is not included in current ad, the returned value is NULL.

- **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 that is not from Facebook doesn't have this. **Usage:** Create a View for AdChoices separately. It is different from Ad corner mark.

Constructor:

```
DUAdChoicesView *choiceView = [[DUAdChoicesView alloc]
initWithNativeAd:nativeAd expandable:NO];
```

Code Sample:

```
[self.adChoicesView addSubview:_chioceView];
```

- **Ad channel type**

Indicate the ad source channel.

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

Description
Indicate the ad source channel.
DUAdChannelTypeUnknow,

DUAdChannelTypeDownload,
DUAdChannelTypeFacebook,

8. 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 Instruction:**

- (1)
- (void)registerViewForInteraction:(nonnull UIView *)view
withViewController:(nullable UIViewController *)viewController;
- or (2)
- (void)registerViewForInteraction:(nonnull UIView *)view
withViewController:(nullable UIViewController *)viewController
withClickableViews:(nullable NSArray<UIView *> *)clickableViews;

Parameters	Description
(UIViewController *)viewController	Clickable View in Ad contents
(NSArray<UIView *> *)clickableViews	More detailed sub-View

- **Code Sample:**

```
//(1)
[nativeAd registerViewForInteraction:self.adUIView withViewController:self];

//(2)
NSArray *clickableViews = @[self.adCallToActionButton, self.adCoverMediaView];
[nativeAd registerViewForInteraction:self.adUIView withViewController:self
withClickableViews:clickableViews];
```

- **Unregister the view:**

- (void)unregisterView;

9. Register interstitial ad



Figure a sample of interstitial ad

9.1 Declaration

In your View Controller header file, import the `DUModuleSDK`, declare that View Controller implements the `DUInterstitialAdDelegate` protocol and add a `DUInterstitialAd` instance variable:

- **Code Sample:**

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

@interface ViewController:UIViewController <DUInterstitialAdDelegate>
@property (strong, nonatomic) DUInterstitialAd * interstitialAd;
@end
```

9.2 Construction

- **Interface Instructions:**

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

Parameters	Description
int pid	DAP placement ID, see 3.2
int cacheSize	ad cache size. Cachesize could be set to 1-5. Recommend not to set cachesize. The default cachesize will be 1.

● **Code Sample:**

```

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

```

9.3 Pre-cache the interstitial ad

● **Interface Instruction:**

- (void)fillAd;

Use the fillAd to pre-cache ads in advance, for faster loading the ads when using loadAd.

Suggestion: Use 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.

9.4 Retrieve interstitial ad

Please register a callback interface for receiving the native ad data. The ad retrieving process is asynchronous, so it will not block developers' threads.

9.4.1 Set delegate for interstitial ad

● **Interface Instruction:**

@protocol DUInterstitialAdDelegate <NSObject>

Description
<p>/* Retrieve ad successfully after calling loadAd */</p> <p>- (void) interstitialAdDidLoad:(nonnull DUInterstitialAd *) interstitialAd;</p>
<p>/* Sent immediately before the impression of an Ad object will be logged*/</p> <p>- (void) interstitialAdWillLogImpression:(nonnull DUInterstitialAd *) interstitialAd;</p>
<p>/* Get an error*/</p> <p>- (void) interstitialAd:(nonnull DUInterstitialAd *) interstitialAd didFailWithError:(nonnull NSError *)error;</p>


```
/* Retrieve an ad click event*/  
- (void) interstitialAdDidClick:(nonnull 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;
```

9.4.2 Retrieve ad

- **Interface Instruction:**

- (void)loadAd;

9.4.3 Show ad

- **Interface Instruction:**

- (BOOL)showAdFromRootViewController:(nullable UIViewController *)
rootViewController;

9.5 Get cached native ad

9.5.1 Check if there is cached ad

- **Interface Instruction:**

- (BOOL)isHasCached;

9.5.2 Get cached native ad

- **Interface Instruction:**

- (DUInterstitialAd *)getCacheAd;

This interface is for getting the cached ad for the current ad object. 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.