Intergration Guide for Cheetah Mobile Ads-iOS-SDK

(Version1.5)

Directory

Prerequisites

- Xcode5.1 or higher
- Deployment target of 7.0 or higher

1. Creating your MID&POSID for integrated application

Please visit http://pub.adkmob.com/index.php to create MID and POSID for your application.

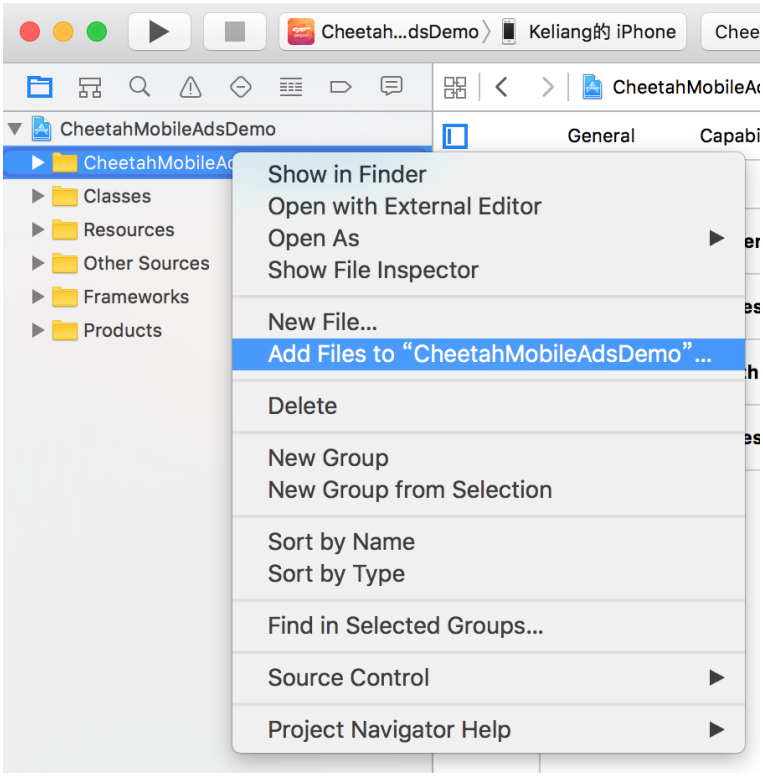
（Please open your account for Supply Side Platform from Business Development Department of Cheetah Mobile at first）.

2. Intergrating the SDK into your Xcode project(manually)

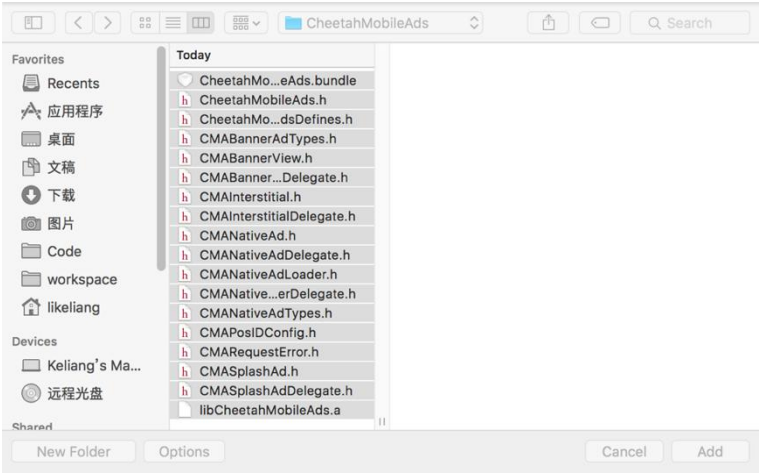
If you do not have CheetahMobileAds downloaded already, grab it from Github（https://github.com/CMAdSDK/pegasi-ios-sdk）and unzip it.

Step 1. Add the SDK to your Xcode project

Right-click on the CheetahMobileAdsDemo project, and choose"Add Files To CheetahMobileAdsDemo" ,to add files to CheetahMobileAdsDemo.



Add the SDK under the CheetahMobileAds file:



Step 2. Add other frames that the SDK required

The SDK depends on the following iOS development frameworks which may not already be part of your project:

- AdSupport
- CoreTelephony

- StoreKit
- CoreLocation
- Security
- SystemConfiguration

Navigate to Project Settings > Build Phases

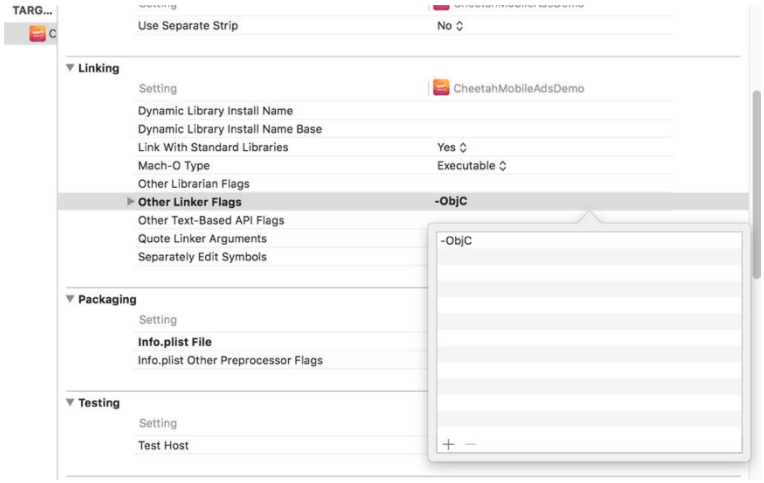
Select "Link Binary with Libraries" Option

Click "+" to add other frameworks required

Step 3. Change settings of static library

Navigate to Target>Build Settings>Linking

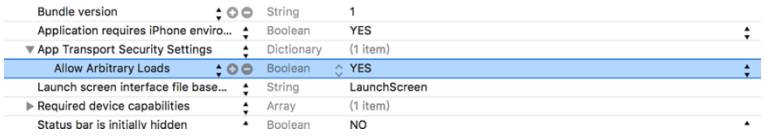
Select“Other Linker Flags”option and add-ObjC（as shown below）



Step 4. Modify AppTransportSecurity settings of info.plist（for iOS 9）

Now that iOS 9 has introduced AppTransportSecurity(ATS) characteristic, Developers are expected to add NSAppTransportSecurity settings in their projects, in order to support http requesting. The detailed settings are as

follows:



3. Initializing CheetahMobileAds

Import CheetahMobileAds in your AppDelegate.h

```
#import "CheetahMobileAds.h"
```

Set MID for CheetahMobileAds when the program starts.

Add code in -application: didFinishLaunchingWithOptions method.

```
[CheetahMobileAds startWithMIDConfig:[[CMAMIDConfig alloc]

initWithMID:@"YOUR_MID" chinaMID:@"YOUR_MID_INCHINA"]];
```

4. Creating a native ad unit

Step 1. Declare CMANativeAdLoader object and CMANativeAd object in NativeAdDetailsViewController

```
@interface NativeAdDetailsViewController ()<CMANativeAdLoaderDelegate>
@property (nonatomic) CMANativeAdLoader *adLoader;
@property (nonatomic) CMANativeAd *nativeAd;
@end
```

Step 2. Initialize CMANativeAdLoader

```
- (void)viewDidLoad {
    [super viewDidLoad];
    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]
initWithPosID:@"1345103" chinaPosID:@"1346103"];
    self.adLoader = [[CMANativeAdLoader alloc]
initWithPosIDConfig:config adTypes:@[ kCMANativeAdLoaderNewsFeed]];
    self.adLoader.delegate = self;
}
```

Step 3. Call the -loadAd method when you expect to load an ad unit

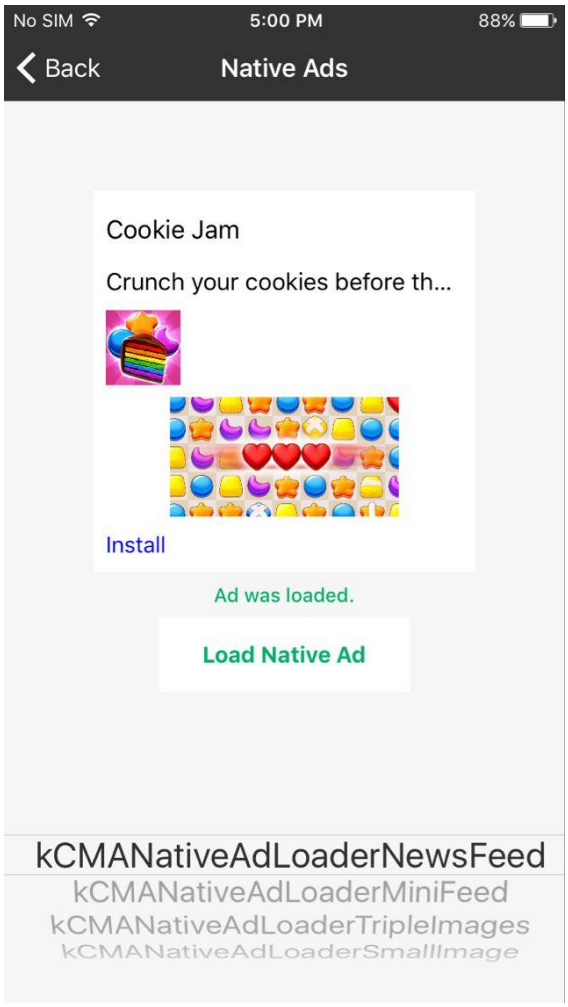
```
[self.adLoader loadAd];
```

Step 4. Implement CMANativeAdLoaderDelegate method to track different life cycle stages for your ad unit (optional)

```
// When the ad loading succeeds, callback:

- (void)nativeAdLoaderLoaded:(CMANativeAdLoader *)nativeAdLoader;

// When the ad loading fails, callback:

- (void)nativeAdLoader:(CMANativeAdLoader *)nativeAdLoader

didFailToReceiveAdWithError:(CMARequestError *)error;
```

Results :

The native ad data showcase in our Ad Demo is displayed as follows:



5. Creating a banner ad unit

A BannerView can be created from Storyboard or from code. This guide shows the code method to create a BannerView with the size of 320*50.

Step 1. Declare CMABannerView object in BannerAdViewController

```
@interface BannerAdViewController ()<CMABannerViewDelegate>
@property (strong, nonatomic) CMABannerView *banner;
@end
```

Step 2. Initialize CMABannerView in -viewDidLoad

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]
initWithPosID:@"1345105" chinaPosID:@"1346105"];
    self.banner = [[CMABannerView alloc]
initWithBannerAdType:kCMABannerAdTypeBanner];
    self.banner.delegate = self;
    self.banner.posIDConfig = config;
    self.banner.rootViewController = self;
}
```

Step 3. Call the -loadAd method to load the ad into your BannerView

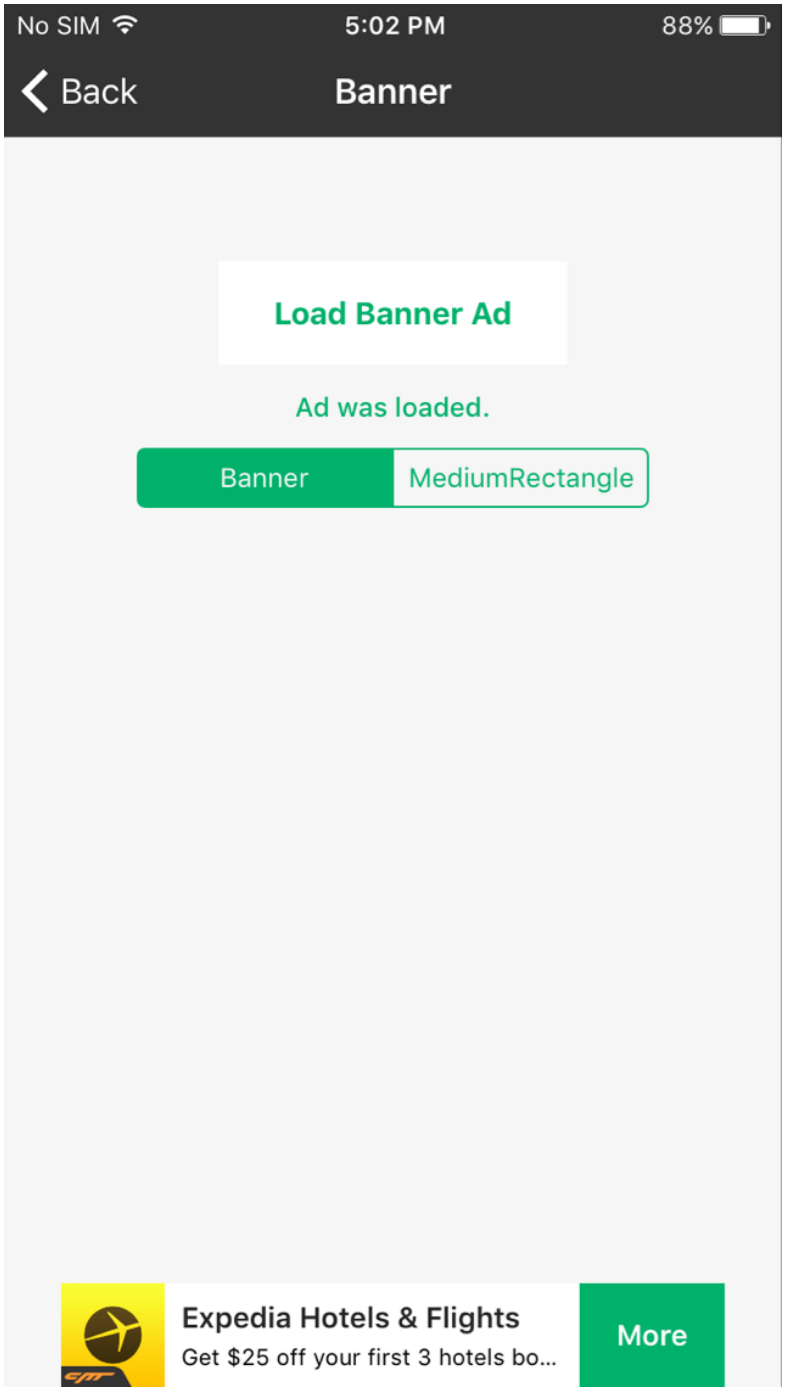
```
- (IBAction)didClickedLoadAdBtn:(id)sender
{
      [self.banner loadAd];
       self.adStatusLabel.text = @"Requesting ad...";
}
```

Step 4. Implement CMANativeAdLoaderDelegate method to track different life cycle stages for your ad unit (optional)

```
// When ad loading succeeds, callback:
- (void)bannerViewDidLoadAd:(CMABannerView *)banner;
// When ad loading fails, callback:
- (void)bannerView:(CMABannerView *)banner
didFailToLoadAdWithError:(CMARequestError *)error;
// (When the ad is clicked,) callback before the display of correspondent
web view:
- (void)bannerViewWillPresentScreen:(CMABannerView *)banner;
// When the displayed web view is closed, callback:
- (void)bannerViewWillDismissScreen:(CMABannerView *)banner;
// (When the ad is clicked,)callback before leaving the application:
- (void)bannerViewWillLeaveApplication:(CMABannerView *)banner;
```

Results：

The banner ad showcase in our Ad Demo is displayed below:



6. Creating an ad unit for interstitial

Step 1. Declare variable CMAInterstitial in AppDelegate

```
@interface InterstitalAdViewController ()<CMAInterstitialDelegate>

@property (nonatomic) CMAInterstitial *interstitial;

@end
```

Step 2. Initialize variable CMAInterstitial in -viewDidLoad

```
- (void)viewDidLoad {

    [super viewDidLoad];

    CMAPosIDConfig *config = [[CMAPosIDConfig alloc]

initWithPosID:@"1345104" chinaPosID:@"1346104"];

    self.interstitial = [[CMAInterstitial alloc] initWithPosIDConfig:config];

    self.interstitial.delegate = self;

}
```

Step 3. Call the -loadAd method when you expect to load an ad unit

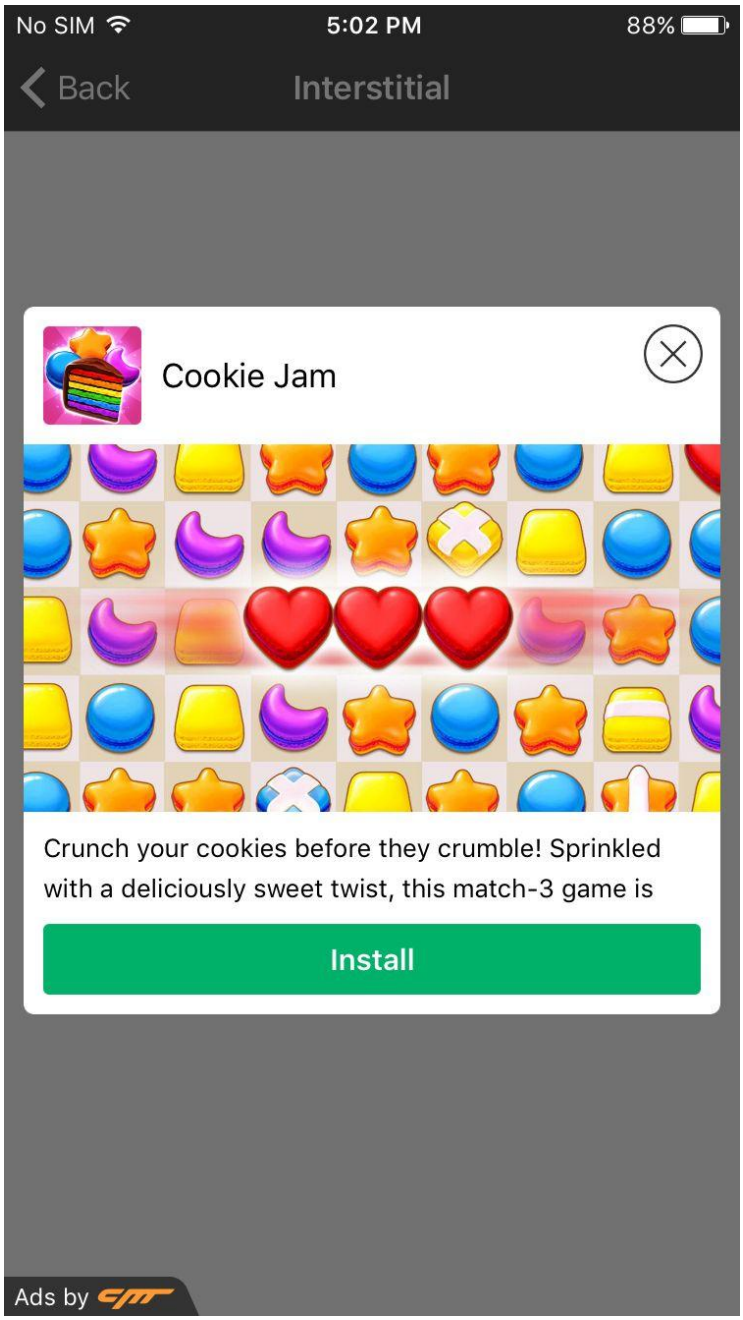
```
- (IBAction)didClickedLoadAdBtn:(id)sender

{

    self.adStatusLabel.text = @"Requesting ad...";

    [self.interstitial loadAd];

}
```

Step 4. Implement CMANativeAdLoaderDelegate method to track different life cycle stages for your ad unit (optional)

```
// When ad loading succeeds, callback:

- (void)interstitialDidLoadAd:(CMAInterstitial *)ad;

//When ad loading fails, callback:

- (void)interstitial:(CMAInterstitial *)ad

didFailToLoadAdWithError:(CMARequestError *)error;

//Before the interstitial ad pops up, callback:

- (void)interstitialWillPresentScreen:(CMAInterstitial *)ad;

//Before the disappear of interstitial ad , callback:

- (void)interstitialWillDismissScreen:(CMAInterstitial *)ad;

//(When the ad is clicked,) callback before leaving the application:

- (void)interstitialWillLeaveApplication:(CMAInterstitial *)ad;
```

Results：

The ad unit showcase for interstitial in our Ad Demo is shown below:



7. Creating an ad unit for splash screen

Step 1. Add code for splash screen in-application:

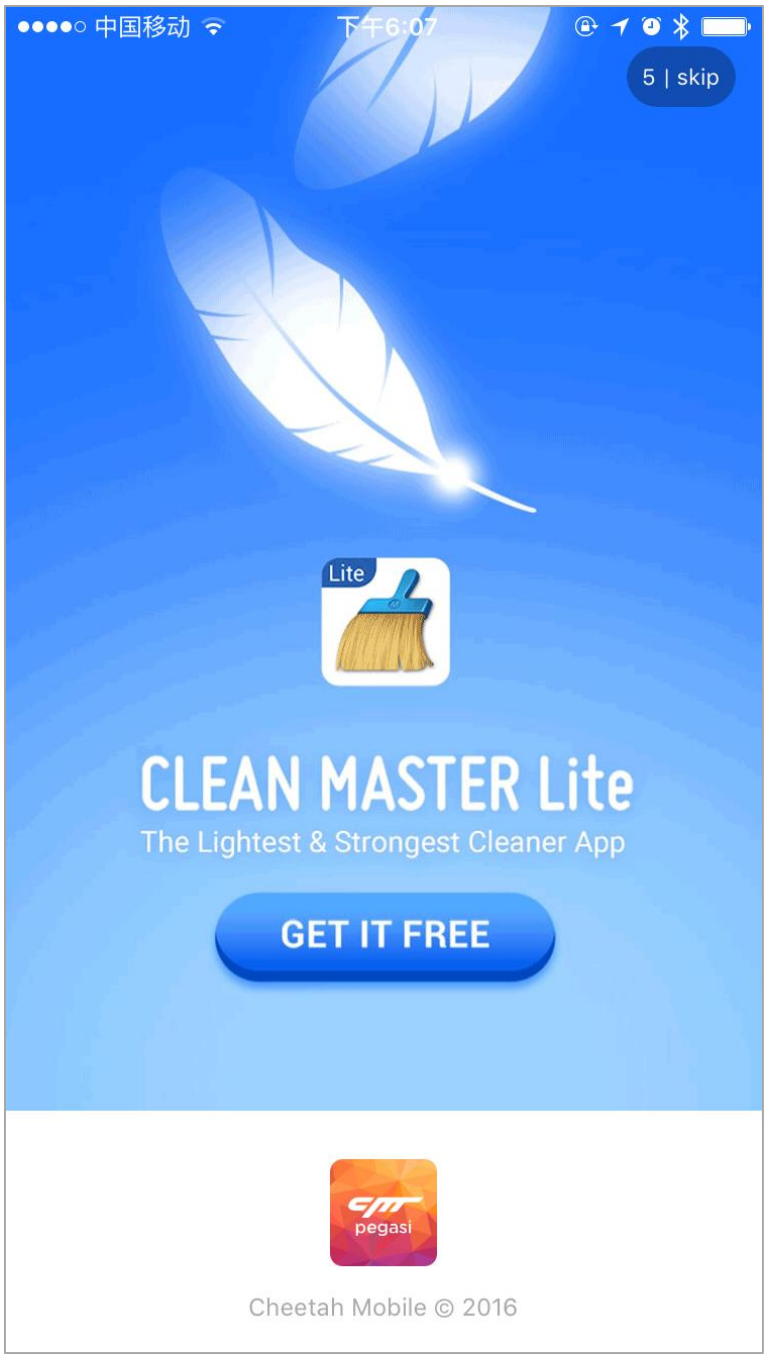
```
didFinishLaunchingWithOptions method

- (BOOL)application:(UIApplication *)application

didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {



      [CheetahMobileAds startWithMIDConfig:[[CMAMIDConfig alloc]

initWithMID:@"1345" chinaMID:@"1346"]];



      CMAPosIDConfig *config = [[CMAPosIDConfig alloc]

initWithPosID:@"1345106" chinaPosID:@"1346106"];



      CMASplashAd *splashAd = [[CMASplashAd alloc]

initWithPosIDConfig:config];

      splashAd.delegate = self;

      splashAd.dismissAnimation = kCMASplashAnimationGrowFade;

      splashAd.backgroundView = [self splashBackgroundView];



      [splashAd show];

}
```

Step 2. Implement CMANativeAdLoaderDelegate method to track different life cycle stages for your ad unit (optional)

```
// When ad loading succeeds, callback:
- (void)splashAdDidLoadAd:(CMASplashAd *)splashAd;
//When ad loading fails, callback:
- (void)splashAd:(CMASplashAd *)splashAd
didFailToLoadAdWithError:(CMARequestError *)error;
//Before the display of splash screen, callback
- (void)splashAdWillPresentScreen:(CMASplashAd *)splashAd;
//Before the disappear of splash screen, call back
- (void)splashAdDidDismissScreen:(CMASplashAd *)splashAd;
// ( When the ad is clicked, ) callback before leaving the application:
- (void)splashAdWillLeaveApplication:(CMASplashAd *)splashAd;
//When the “skip” button is clicked, callback:
- (void)splashAdWillSkip:(CMASplashAd *)splashAd;
```

Results：

The showcase of ad unit for splash screen in our Ad Demo is given below:



8. Error Code Description

-1001	Internal error
-1002	Ad loading timeout
-1003	Network environment error
-1004	No ad filling error
-1005	JSON parsing error
-1006	Sever error
-1007	Wrong requesting parameters
-1008	Resource file downloading error