Dremobi SDK Documentation

Dreamobi Technologies co.,Ltd.
Author: Nicky

Email: nicky.wei@dreamobi.com

Catalog

1. Dreamobi SDK Introduction

Dreamobi SDK provide package of tool to developer for development of HD and Instant-play in-app video ads.

We support 3 format of ads in our SDK currently: Video Wall, Click to Play and Native ads, which could provide high eCPM in industry.

2. Before integration

2.1 Info required from developer

1. Download link of your application
2. Bundle_id of your application
3. Formats of ads willing to integrate : a. Video Wall b. Click to play c. NativeAd

3. Process of integration

3.1 import Dreamobi SDK to the project

1. Import Dreamobi SDK to Project
2. Add Framework of system
   MediaPlayer Framework
   SystemConfiguration Framework
   AdSupport.framework

3.2 Initialization under AppDelegate.m

1. Add Header of Dreamobi SDK
   #import <DreamobiSDK/DreamobiConfigure.h>

2. Initialization function

```
[[DreamobiConfigure config] configureWithAppToken:@"token" zoneIDs:@[@"",@"",@""] appBundleId:@"bundle_id" delegate:self];
```

Explanation of the parameters

App_token	For the validation of authority
Bundle_id	For bundle_id of App store

zoneIDs	For ad Position Id of ads(In the way of array, must have)

3. Follow the protocol of DreamobiAdVideoRewardDelegate to realize the proxy method

```
#import "AppDelegate.h"
#import <DreamobiSDK/DreamobiConfigure.h>

@interface AppDelegate ()<DreamobiAdVideoRewardDelegate>

@end
```

4. The callback logic for implementation of WallAd and VideoAd display(Logic self-defined by developer)

Introduction of function: Dreamobi SDK could have different status after user watched the video and developer could implement the different processing logic with the status, like "Increase the coin of user" for example.

```
-(void)dreamobiAdVideoReward:(BOOL)success isConnect:(BOOL)isConnect  getPoint:(long)point
{
    if (success) {
        NSLog(@"Video play success get point%ld",point);
    }
    if (isConnect) {
        NSLog(@"Play video network can support");
    }else{
        NSLog(@"Play video network can't support");
    }
}
```

Explanation of the parameter:

Success	Identification for the video been played successfully or not
isConnect	Identify if the user connect to Internet
point	Amount of coin grant to user

3.3 Implementation of Video Wall(Optional)

1. To add the header in the target interface that developer want to display
   #import <DreamobiSDK/DreamobiConfigure.h>
2. To add the following method in the event of "click the button to display the video wall"

```
[DreamobiConfigure showAdWallWithZone:@"" withDelegate:self];
```

Note: Developer could convey "nil" in above method if you don't need the status of Start & End of the video and ignore the step3 below and zoneId for the showAdWallForZone method equivalent to the zoneID for the format of ads in Configure.

```
[DreamobiConfigure showAdWallWithZone:@"" withDelegate:nil];
```

3. Follow the protocol of DreamobiAdDelegate (You need to follow this step if the convey is not "nil" in step 2)
   The button of ViewController need to realize the protocol of DreamobiAdDelegate

```
#import "ViewController.h"
#import <DreamobiSDK/DreamobiConfigure.h>

@interface ViewController ()<DreamobiAdDelegate>
```

A. Implementation the callback of video play start

```
- (void)dreamobiAdVideoStarted
{
    NSLog(@"video started ");
}
```

B. Implementation the callback of video play end

```
|
- (void)dreamobiAdVideoFinished
{
    NSLog(@"video Finished ");
}
```

3.4 Implementation of Click to Play(Optional)

1. To add the header below at the interface of Button click to play
2. Determine whether there is can play the video

```
[DreamobiConfigure isAvailVideoWithZoneID:@""];
```

3. Implementation the method in the event of click the button to play the video

```
[DreamobiConfigure showVideoWithZone:@"" withDelegate:self];
```

Note: Developer could convey "nil" in above method if you don't need the status of Start & End of the video and zoneId for showVideoAdForZone method equivalent to the zoneID for the format of ads in Configure.
Follow DreamobiAdDelegate protocol, reference3.3-3

4. This function is for the cache of video on device and imply the following method to get the information of cached video

```
DreamobiVideo *videoMessage = [[DreamobiVideo alloc] init];
videoMessage = [DreamobiConfigure getVideoMessageWithZoneID:@""];
```

zoneID for getVideoMessageWithZoneID method equivalent to the zoneID of video need click to play
You could get the target of DreamobiVideo to execute this method with the related info of video.

3.5 Implementation of NativeAd(Optional)

1. Method to implement the request for NativeAdMessage

    ```
    [DreamobiConfigure getNativeAdWithZone:@"|" withIconSize:NOSIZE withBannerSize:NOSIZE withDelegate:self];
    ```

    zoneId equivalent to the zoneID for the format of ads

    withDelegate convey self to execute the callback to obtain NativeAd (Need to follow the protocol in step 2)

    It will not execute the callback if **withDelegate** is nil

    IconSize & BannerSize don't need size requirement, convey NOSIZE

    If size required, the specification as below:

    ```
    |
    NOSIZE=0,

    ICON_50X50 =1,
    ICON_100X100 =2,
    ICON_175X175 =3,
    BANNER_1200X627 =4,
    BANNER_640X334 =5,
    ```

2. The request of callback for NativeAd need to follow the protocol of DreamobiNativeAd (for the callback of nativead)

    ```
    @interface ViewController ()<DreamobiNativeAd>
    ```

    The method callback need to be implemented

    ```
    - (void)dreamobiNativeAd:(DreamobiNative *)nativeMessage
    {
        NSLog(@"native-----%@",nativeMessage);
    }
    ```

    The callback is the target of DreamobiNative and the attribute is the info of nativeAd related.
    The target of callback is blank if no data been requested.