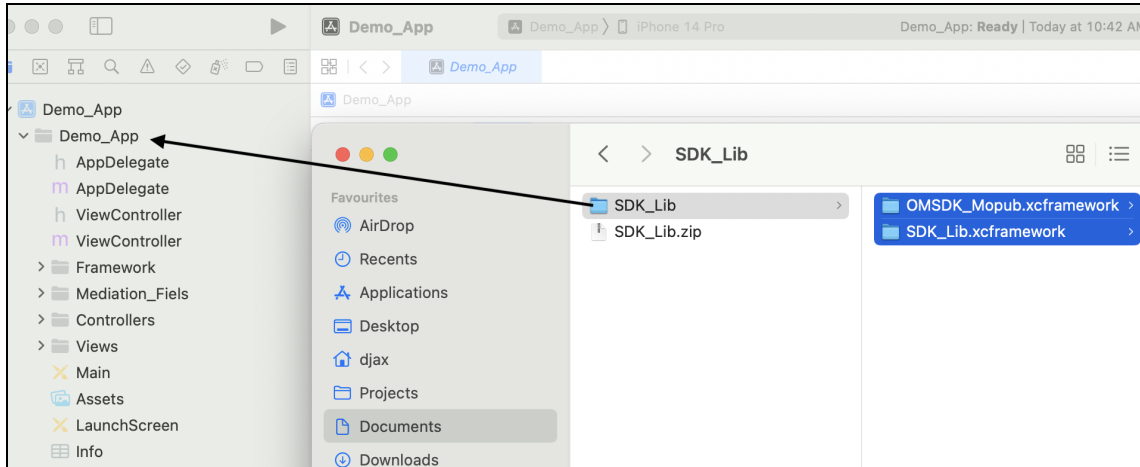


iOS AD FORMATS INTEGRATION GUIDE

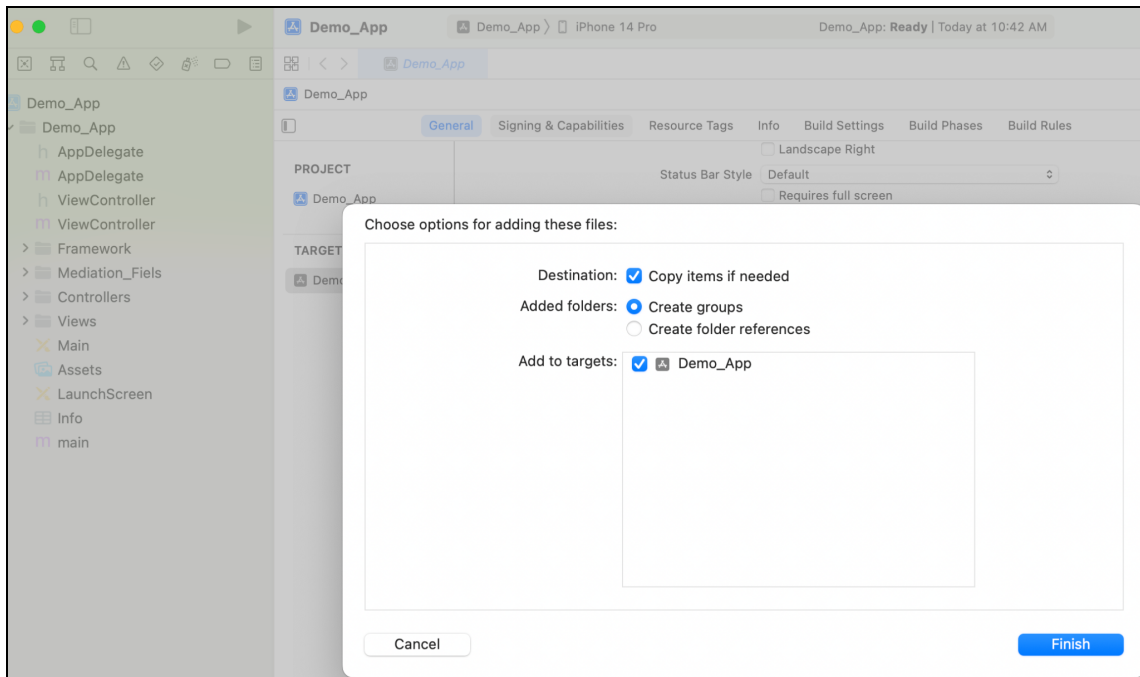
September 2022

Steps to integrate Xcframework

1. To extract the SDK_Lib.zip file, drag and drop the SDK_Lib.xcframework and OMSDK_Mopub.xcframework into the project.



2. Then, make sure the following options are selected for adding files. Both “Copy items if needed” and “Create groups” should be checked and selected. After clicking the “Finish” button.

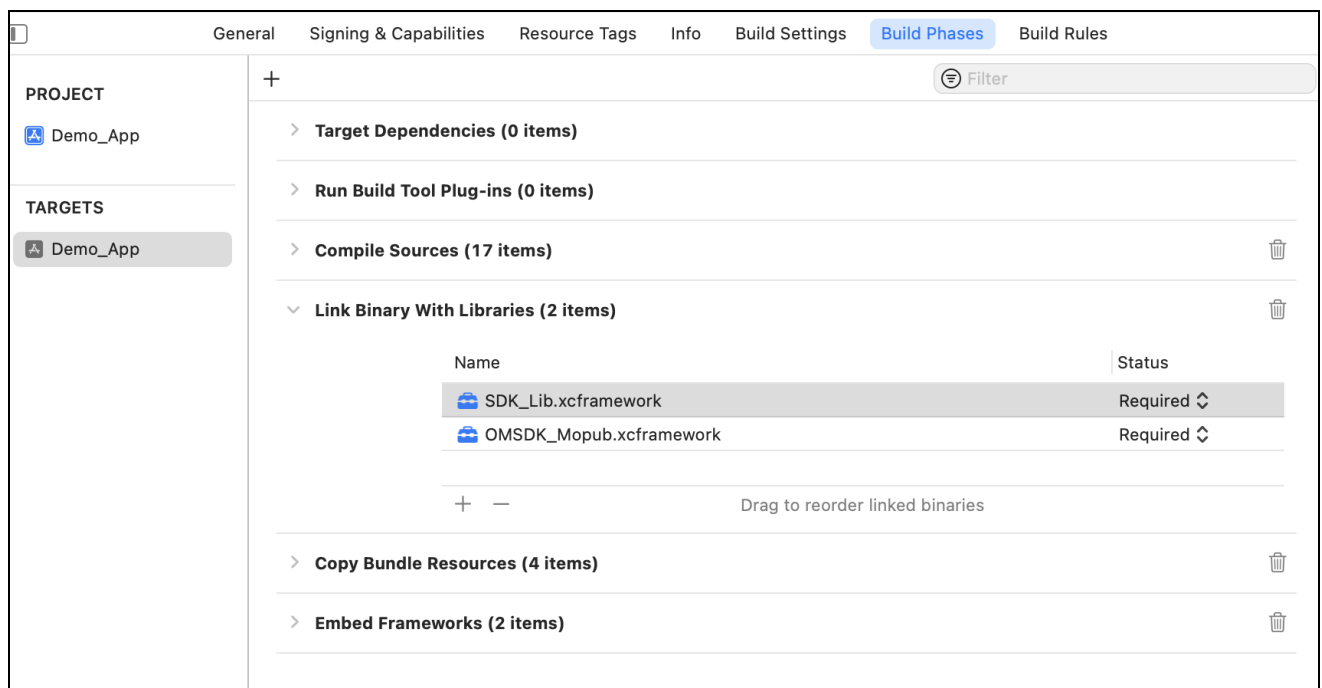


3. In order to make sure that the framework will get copied the app's binary, follow these steps:

- a). Navigate to project settings by clicking on it in the project navigator.
- b). Make sure that project target is selected and the General tab is open.
- c). Select “Embed & Sign” for newly added XCFramework.

4. XCFramework Link with project

Navigate to the Build Phases tab, disclose the “Link Binary With Libraries” list and make sure the framework is included in the list. It should already be included by default after following the steps above, however in case it's not – click on the + button and add it.



The XCFramework is now fully added and integrated with the Xcode project.

Prerequisites

1. Info plist add the App Transport Security Settings → Allow Arbitrary Loads → YES

Quick Guide

Below the steps support the Internal ads (Banner, Interstitial Image, Interstitial Video, Rewarded Video, Native Image) and Geofence Interstitial image ad, Geofence Text ad and Mediation ads (Banner, Interstitial Image, Interstitial Video, Rewarded Video, Native Video) also.

Note : Ad does not show if any integration code sets are missed.

Banner Ad

1. **Get Banner Ad zone ID from Digiads Publisher Login (Web Portal).**

The zone id that would be configured to deliver Banner Ad into the app. This will be linked to the publisher's ID and can be obtained from Digiads Publisher Login.

2. **Add the code in the class file to invoke the Banner Ad view.**

This is the Objective C code which needs to be placed in the classes where banner ads need to be invoked.

Code Snippets

The class Ad_Response is used to bring in banner ads to the app. It can be placed anywhere in the app like a normal banner ad. It usually appears at the top or bottom of the app's screen. These integration steps included banner ads.

Banner Ad - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Banner Ad - Method

```
Ad_Response *response = [[Ad_Response alloc] init];  
[self.view addSubview:[response basic_ad_PassZoneid:zoneid Position:@"Top" PassCoordinateXAxis:0  
PassCoordinateYAxis:10 PassFrameWidth:UIScreen.mainScreen.bounds.size.width  
PassFrameHeight:UIScreen.mainScreen.bounds.size.height]];
```

Interstitial Image Ad

1. **Get Interstitial Image Ad zone ID from Digiads Publisher Login (Web Portal).**

The zone id that would be configured to deliver Interstitial Image Ad into the app. This will be linked to the publisher's ID and can be obtained from Digiads Publisher Login.

2. **Add the code in the class file to invoke the Interstitial Image Ad view.**

This is the Objective C code which needs to be placed in the classes where Interstitial Image ad needs to be invoked.

Code Snippets

The class Ad_Response is used to bring in the Interstitial Image ads to the app. It can be placed anywhere in the app like a normal Interstitial Image ad. This ad will appear on the whole screen with a close button.

Interstitial Image Ad - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Interstitial Image Ad - Method

```
Ad_Response *response = [[Ad_Response alloc] init];  
[self.view addSubview:[response intImg_ad_PassZoneid:zoneId]];
```

Interstitial Video Ad

Video Ad format is a full screen video ad which publishers can make use of to show any video related ads to the app. Interstitial ads provide full-screen experiences, commonly incorporating rich media to offer a higher level of interactivity compared to other ads. Interstitials are typically shown during natural transitions in the app, for example, after completing a game level, or while waiting for a new view to load.

1. **Get Video Ad zone ID from Digiads Publisher Login (Web Portal).**

The zone id that would be configured to deliver Video ads into the app. This will be linked to publisher ID and can be obtained from Publisher login.

2. **Add the Objective C code in the class file to invoke the video ad view**

This is the objective c code which needs to be placed in the classes where video ad needs to be invoked. This will not only set the view but various other parameters like zone ID. This also consists of various callbacks that can be added which the publisher can use to view various parameters that SDK gives to publishers to make effective use of.

Code Snippets

The class Ad_Response is used to bring in the Interstitial video ads to the app. It can be placed anywhere in the app like a normal Interstitial video ad.

Interstitial Ad - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Interstitial Ad - Method

```
Ad_Response *response;  
response = [[Ad_Response alloc] init];  
[self.view addSubview:[res intVid_ad_PassZoneid:globalzonevalue]];
```

Rewarded Video Ad

Rewarded video advertising is a format that gives users a reward in exchange for time spent viewing a full-screen ad. Rewarded videos are typically between 15-30 seconds in length and cannot be skipped.

Rewarded ads are a great way to keep users engaged in the app while earning ad revenue. The reward generally comes in the form of in-game currency (gold, coins, power-ups, etc.) and is distributed to the user after a successful ad completion.

1. Get Video Ad zone ID from Digiads Publisher Login (Web Portal).

The zone id that would be configured to deliver Rewarded video ads into the app. This will be linked to the publisher's ID and can be obtained from Publisher login. Once Rewarded video ad completed users get the reward points.

2. Add the code in the class file to invoke the Video Ad view.

This is the objective c code which needs to be placed in the classes where rewarded ad needs to be invoked.

Code Snippets

The class Ad_Response is used to bring in the Interstitial video ads to the app. It can be placed anywhere in the app like a normal Rewarded video ad.

Rewarded video Ad - Header

```
#import <SDK_Lib/Ad_Response.h>  
#import <SDK_Lib/MPRewardedAds.h>  
@interface ViewController() <MPRewardedAdsDelegate>
```

Rewarded video Ad - Method

```
Ad_Response *res1 = [[Ad_Response alloc] init];
[MPRewardedAds setRewardDelegate:self];
[self.view addSubview:[res1 PassZoneid:zoneId isRewardDelegate:@"TRUE"]];
```

Rewarded Video Ad [Delegate] - Method

In the rewarded ad delegate methods it will show the ad stages like - Did load, Present reward ad, Fail to load, Fail to show, Will Present, Did Present, Will Dismiss, Did Dismiss, Did Expire, Receive tap event, Leave application and after some seconds the reward points will deliver in the app side.

```
- (void)rewardedAdDidLoadForAdZoneID:(NSString *)adZoneID {
[MPRewardedAds presentRewardedAdForAdZoneID:zoneId fromViewController:self withReward:nil
customData:@"testCustomerId"]; }
- (void)rewardedAdDidFailToLoadForAdZoneID:(NSString *)adZoneID error:(NSError *)error {}
- (void)rewardedAdDidFailToShowForAdZoneID:(NSString *)adZoneID error:(NSError *)error {}
// Called when there is an error while attempting to show the ad.
- (void)rewardedAdWillPresentForAdZoneID:(NSString *)adZoneID {}
// Called when a rewarded ad starts playing.
- (void)rewardedAdDidPresentForAdZoneID:(NSString *)adZoneID {}
- (void)rewardedAdWillDismissForAdZoneID:(NSString *)adZoneID {}
- (void)rewardedAdDidDismissForAdZoneID:(NSString *)adZoneID {}
// Called when a rewarded ad is completed and the user should be rewarded.
- (void)rewardedAdDidExpireForAdZoneID:(NSString *)adZoneID {}
// Called when a rewarded ad is expired.
- (void)rewardedAdDidReceiveTapEventForAdZoneID:(NSString *)adZoneID {}
- (void)rewardedAdWillLeaveApplicationForAdZoneID:(NSString *)adZoneID {}
// Called when a rewarded ad is closed. At this point the application should resume.
- (void)rewardedAdShouldRewardForAdZoneID:(NSString *)adZoneID reward:(MPReward *)reward {
    NSArray * rewards = [MPRewardedAds availableRewardsForAdZoneID:adZoneID];
    NSLog(@"Rewarded video should reward for ad %@",rewards); }
```

Native Image Ad:

This ad supports the native image ad (use tableview). To import the library file.

```
#import "AdPlacerView.h"
#import <SDK_Lib/MPNativeAdRequestTargeting.h>
#import <SDK_Lib/MPTableViewAdPlacer.h>
#import <SDK_Lib/MPCClientAdPositioning.h>
#import <SDK_Lib/MPNativeAdConstants.h>
#import <SDK_Lib/MPStaticNativeAdRendererSettings.h>
#import <SDK_Lib/MPStaticNativeAdRenderer.h>
#import <SDK_Lib/MPNativeAdRendererConfiguration.h>
#import <SDK_Lib/sdk_lib_new.h>
```

In Interface file header to add these delegates and implement these steps

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

In Implementation File

```
@property (nonatomic) MPTableViewAdPlacer *placer;
```

Get adformats values by using the below code

```
MPNativeAdRequestTargeting *targeting = [MPNativeAdRequestTargeting targeting];
targeting.desiredAssets = [NSSet setWithObjects:kAdIconImageKey, kAdMainImageKey, kAdCTATextKey,
kAdTextKey, kAdTitleKey, nil];
MPStaticNativeAdRendererSettings *settings = [[MPStaticNativeAdRendererSettings alloc] init];
settings.renderingViewClass = [AdPlacerView class];
settings.viewSizeHandler = ^(CGFloat maximumWidth) {
return CGSizeMake(maximumWidth, 312.0f);};
MPNativeAdRendererConfiguration *nativeAdConfig = [MPStaticNativeAdRenderer
rendererConfigurationWithRendererSettings:settings];
MPClientAdPositioning *positioning = [MPClientAdPositioning positioning];
[positioning addFixedIndexPath:[NSIndexPath indexPathForRow:1 inSection:0]];
[positioning enableRepeatingPositionsWithInterval:5];
self.placer = [MPTableViewAdPlacer placerWithTableView:self.tableView viewController:self
adPositioning:positioning rendererConfigurations:@[nativeAdConfig]];
self.placer.delegate = self;
[self.placer loadAdsForAdZoneID:globalzoneee];});
```

Native ad placer delegate

```
- (void)nativeAdWillPresentModalForTableViewAdPlacer:(MPTableViewAdPlacer *)placer{ }
- (void)nativeAdDidDismissModalForTableViewAdPlacer:(MPTableViewAdPlacer *)placer{ }
- (void)nativeAdWillLeaveApplicationFromTableViewAdPlacer:(MPTableViewAdPlacer *)placer{ }
```


Geofence Ad

It is a type of location-based ad that connects with smartphone users in a designated geographic area, such as mobile apps.

1. Get Geofence Ad zone ID from Digiads Publisher Login (Web Portal).

The zone id that would be configured to deliver Geofence ads into the app. This will be linked to the publisher's ID and can be obtained from Digiads Publisher Login.

2. Add the code in the class file to invoke the Geofence ad view.

This is the Objective C code which needs to be placed in the classes where banner ads need to be invoked.

Code Snippets

The class Ad_Response is used to bring in geofence ads to the app. It can be placed anywhere in the app like a normal geofence ad.

Geofence Ads - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Geofence Ads (Interstitial Image Ad & Text Notification Ad)

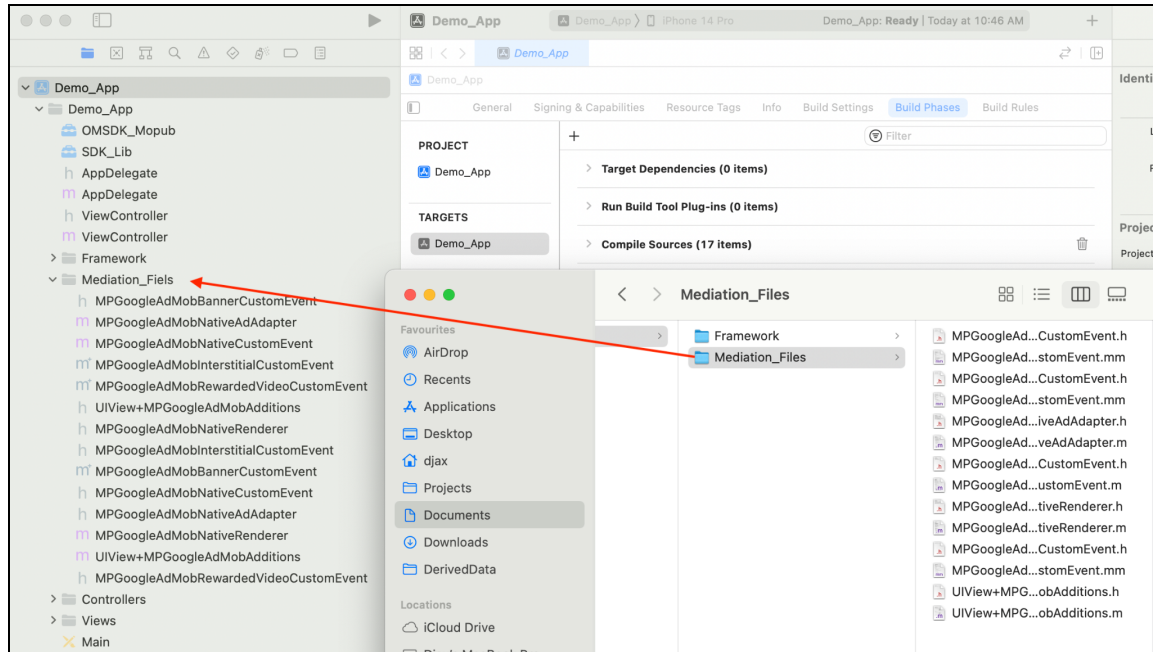
```
Ad_Response *response = [[Ad_Response alloc] init];  
[self.view addSubview:[response basic_ad_PassZoneid:zoneid Position:@"Top" PassCoordinateXAxis:0  
PassCoordinateYAxis:10 PassFrameWidth:UIScreen.mainScreen.bounds.size.width  
PassFrameHeight:UIScreen.mainScreen.bounds.size.height]];
```

Mediation Ad

Publishers want to use the AdMob SDK to load and display ads from SDK via mediation.

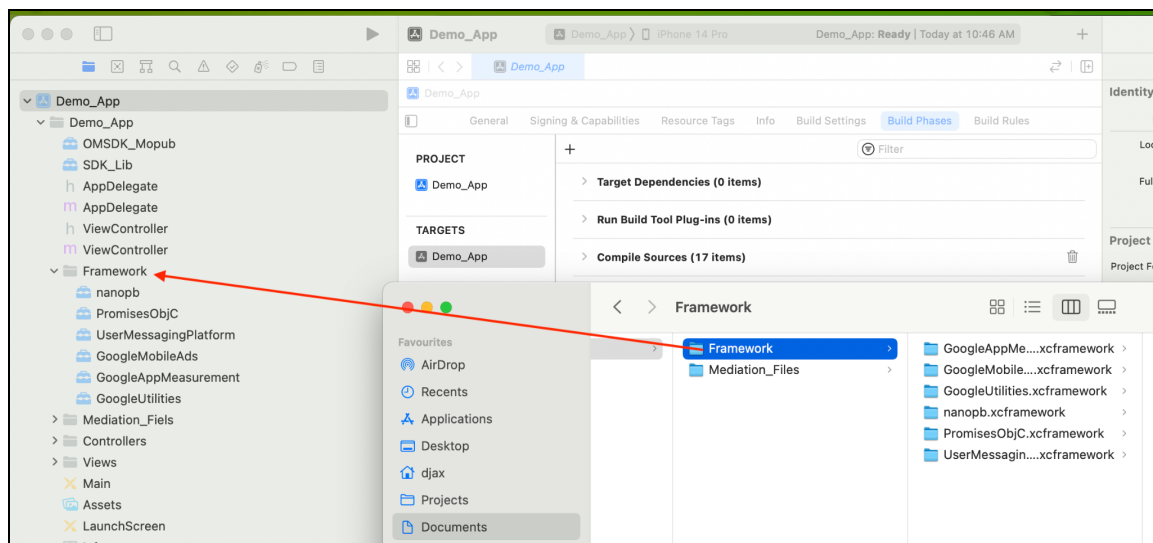
Steps to integrate Mediation Adapter Files

1. To extract the folder, drag and drop the mediation adapter files into the project.



Steps to integrate framework

1. To extract the folder, drag and drop the mediation framework into the project.



Prerequisites :

1. Info.plist add the following steps in demo app
 - GADApplicationIdentifier - ca-app-pub-3940256099942544~1458002511
 - GADIsAdManagerApp - true
 - SKAdnetwork related files added in the demo app, check and add those keys in the demo app.

Banner Ad

1. **Get Banner Ad zone ID from Digiads Publisher Login(Web Portal).**

The zone id that would be configured to deliver Banner ads into the app. To create with the ad unit id in the publisher login.

2. **Add the code in the class file to invoke the Banner Ad view.**

This is the Objective C code which needs to be placed in the classes where banner ads need to be invoked.

Code Snippets

The class Ad_Response is used to bring in banner ads to the app. It can be placed anywhere in the app like a normal banner ad. It usually appears at the top or bottom of the app's screen. These integration steps included banner ads.

Banner Ad - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Banner Ad - Method

```
Ad_Response *response = [[Ad_Response alloc] init];  
[self.view addSubview:[response basic_ad_PassZoneid:zoneid Position:@"Top" PassCoordinateXAxis:0  
PassCoordinateYAxis:10 PassFrameWidth:[UIScreen mainScreen].bounds.size.width  
PassFrameHeight:[UIScreen mainScreen].bounds.size.height]];
```

Interstitial Image Ad

1. Get Interstitial Image Ad zone ID from Digiads Publisher Login(Web Portal).

The zone id that would be configured to deliver Banner ads into the app. To create with the ad unit id in the publisher login.

2. Add the code in the class file to invoke the Interstitial Image Ad view.

This is the Objective C code which needs to be placed in the classes where Interstitial Image ad needs to be invoked.

Code Snippets

The class Ad_Response is used to bring in the Interstitial Image ads to the app. It can be placed anywhere in the app like a normal Interstitial Image ad..

Interstitial Image Ad - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Interstitial Image Ad - Method

```
Ad_Response *response = [[Ad_Response alloc] init];  
[self.view addSubview:[response intImg_ad_PassZoneid:zoneId]];
```

Interstitial Video Ad

Video Ad format is a full screen video ad which publishers can make use of to show any video related ads to the user. Interstitial ads provide full-screen experiences, commonly incorporating rich media to offer a higher level of interactivity compared to other ads. Interstitials are typically shown during natural transitions in the app, for example, after completing a game level, or while waiting for a new view to load.

1. Get Video ad zone ID from Digiads Publisher Login(Web Portal).

The zone id that would be configured to deliver Banner ads into the app. To create with the ad unit id in the publisher login.

2. Add the Objective C code in the class file to invoke the Video Ad view

This is the objective c code which needs to be placed in the classes where Video ad needs to be invoked.

Code Snippets

The class Ad_Response is used to bring in the Interstitial video ads to the app. It can be placed anywhere in the app like a Interstitial video ad.

Interstitial Ad - Header

```
#import <SDK_Lib/Ad_Response.h>
```

Interstitial Ad - Method

```
Ad_Response *response;  
response = [[Ad_Response alloc] init];  
[self.view addSubview:[res intVid_ad_PassZoneid:globalzonevalue]];
```

Rewarded video

Rewarded video advertising is a format that gives users a reward in exchange for time spent viewing a full-screen ad. Rewarded videos are typically between 15-30 seconds in length and cannot be skipped.

Rewarded ads are a great way to keep users engaged in the app while earning ad revenue. The reward generally comes in the form of in-game currency (gold, coins, power-ups, etc.) and is distributed to the user after a successful ad completion.

1. Get Video Ad zone ID from Digiads Publisher Login(Web Portal).

This is the get the zone id that would be configured to deliver rewarded video ads into the app. Once rewarded video ad completed users get the reward points.

2. Add the code in the class file to invoke the Video Ad view.

This is the objective c code which needs to be placed in the classes where banner ad needs to be invoked.

Code Snippets

Rewarded Video Ad - Header

```
#import <SDK_Lib/Ad_Response.h>  
@interface ViewController()
```

Rewarded Video Ad - Method

```
Ad_Response *res1 = [[Ad_Response alloc] init];  
[self.view addSubview:[res1 PassZoneid:zoneId isRewardDelegate:@"TRUE"]];
```

Native video

This ad supports the native video ad. To import the library file.

```
#import "AdPlacerView.h"  
#import <SDK_Lib/MPNativeAdRequestTargeting.h>  
#import <SDK_Lib/MPTableViewAdPlacer.h>  
#import <SDK_Lib/MPClientAdPositioning.h>  
#import <SDK_Lib/MPNativeAdConstants.h>  
#import <SDK_Lib/MPStaticNativeAdRendererSettings.h>  
#import <SDK_Lib/MPStaticNativeAdRenderer.h>  
#import <SDK_Lib/MPNativeAdRendererConfiguration.h>  
#import <SDK_Lib/sdk_lib_new.h>
```

In Interface file header to add these delegates and implement these steps

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

In Implementation File

```
@property (nonatomic) MPTableViewAdPlacer *placer;
```

Get adformats values by using the below code

```
dispatch_async(dispatch_get_main_queue(), ^{  
    MPNativeAdRequestTargeting *targeting = [MPNativeAdRequestTargeting targeting];  
    targeting.desiredAssets = [NSSet setWithObjects:kAdIconImageKey, kAdMainImageKey, kAdCTATextKey,  
    kAdTextKey, kAdTitleKey, nil];  
    MPStaticNativeAdRendererSettings *settings = [[MPStaticNativeAdRendererSettings alloc] init];  
    settings.renderingViewClass = [NativeAdView class];  
    settings.viewSizeHandler = ^(CGFloat maximumWidth)  
    {  
        return CGSizeMake(maximumWidth, 312.0f);  
    }  
};
```

```

};
MPNativeAdRendererConfiguration*nativeAdConfig=[MPStaticNativeAdRenderer
rendererConfigurationWithRendererSettings:settings];
nativeAdConfig.supportedCustomEvents=@[@"MPMPNativeCustomEvent",
@"FlurryNativeCustomEvent",@"MPGoogleAdMobNativeCustomEvent"];
MPClientAdPositioning *positioning = [MPClientAdPositioning positioning];
[positioning addFixedIndexPath:[NSIndexPath indexPathForRow:1 inSection:0]];
[positioning enableRepeatingPositionsWithInterval:5];
self.placer = [MPTableViewAdPlacer placerWithTableView:self.tableView viewController:self
adPositioning:positioning
rendererConfigurations:@[nativeAdConfig]];
self.placer.delegate = self;
[self.placer loadAdsForAdZoneID:globalzonee];
});

```

Test Procedure - iOS

AD tested in this url :

https://digiads.com/api/request_sdk.php?zoneid=1200&mediationid=0&NOTINid=null&is_adtype=0&systemtype=ios

Supported ad format :

1. Internal ad
 - a). Banner
 - b). Interstitial Image
 - c). Interstitial Video
 - d). Rewarded Video
 - e). Native Image
2. Geofence ad
 - a). Geo Text Ad (Notification)
 - b). Geo Interstitial Image
3. Mediation ad
 - a). Banner
 - b). Interstitial Image
 - c). Interstitial Video
 - d). Rewarded Video
 - e). Native Video

1. Internal ad

Banner Ad :

1. Select the Basic ad from the list and enter the zone id, the banner will appear on top or bottom of the screen.

Basic ads include (Banner ads displaying in the same UI)

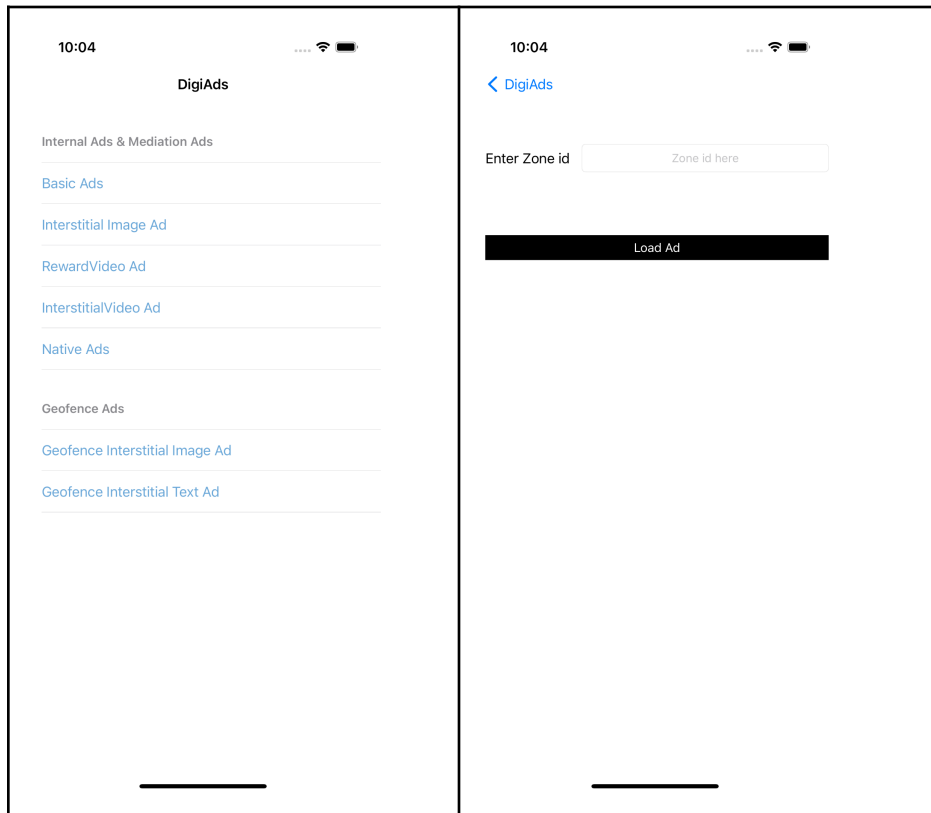


Fig1: Common List Page

Banner ad

10:07



[Back](#)



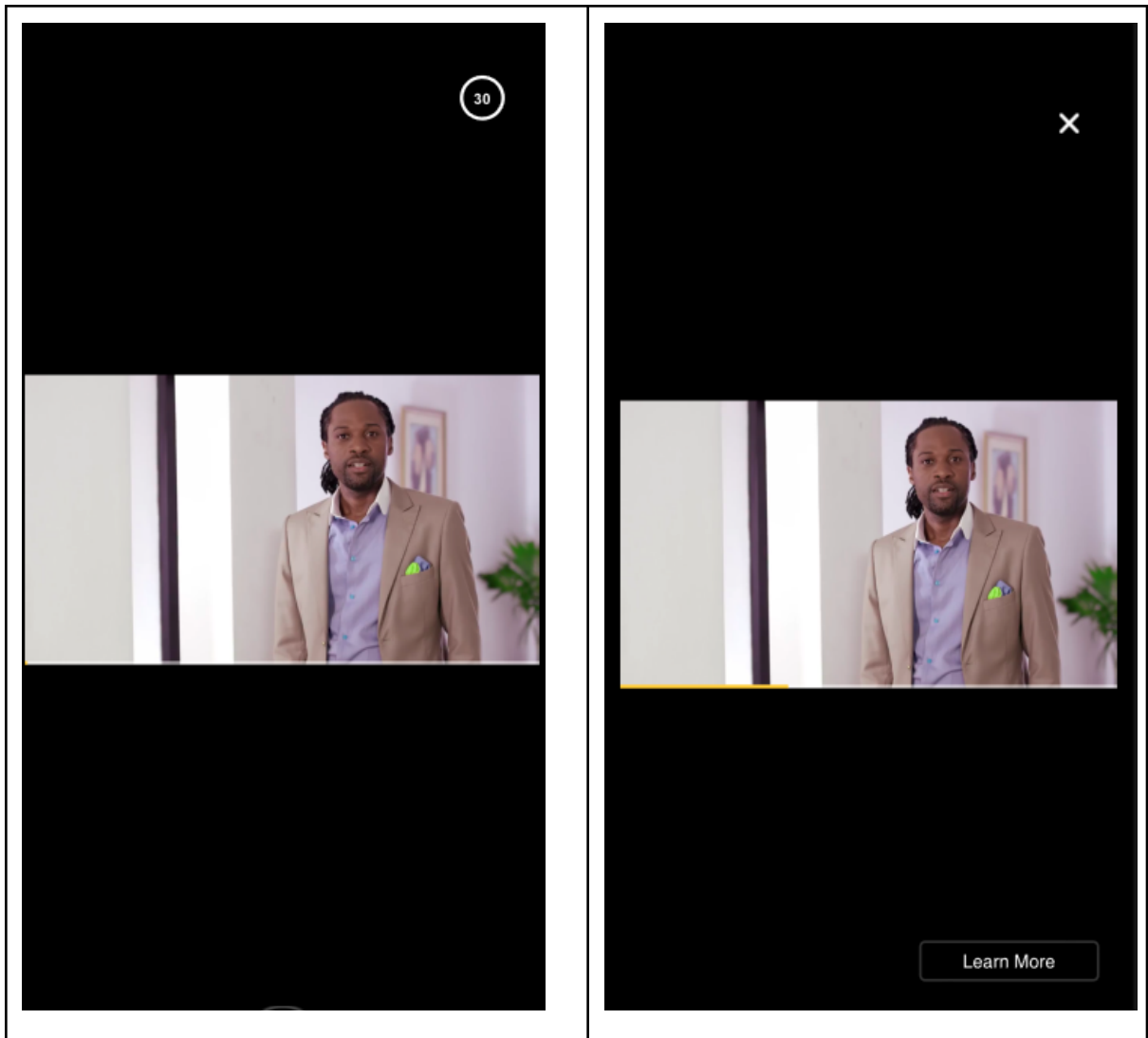
Interstitial Image Ad :

1. Select the Interstitial image ad from the list and enter the zone id (**Ref Fig:1**), the interstitial image will appear full screen.



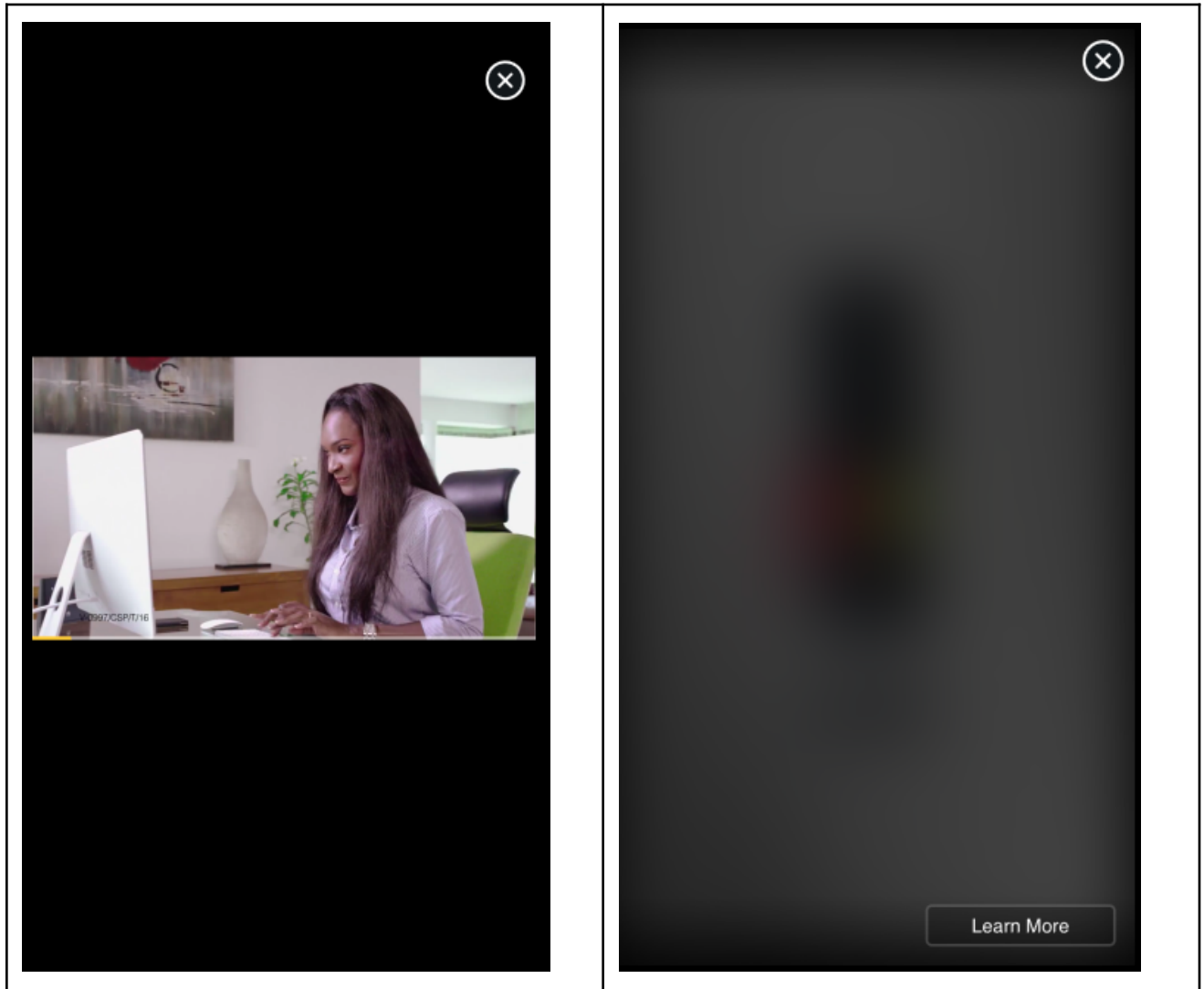
Rewarded Video Ad :

1. Select the rewarded video ad from the list and enter the zone id (**Ref Fig:1**), the rewarded video playing in full screen with a timer, once the ad is finished the user will receive the reward points.



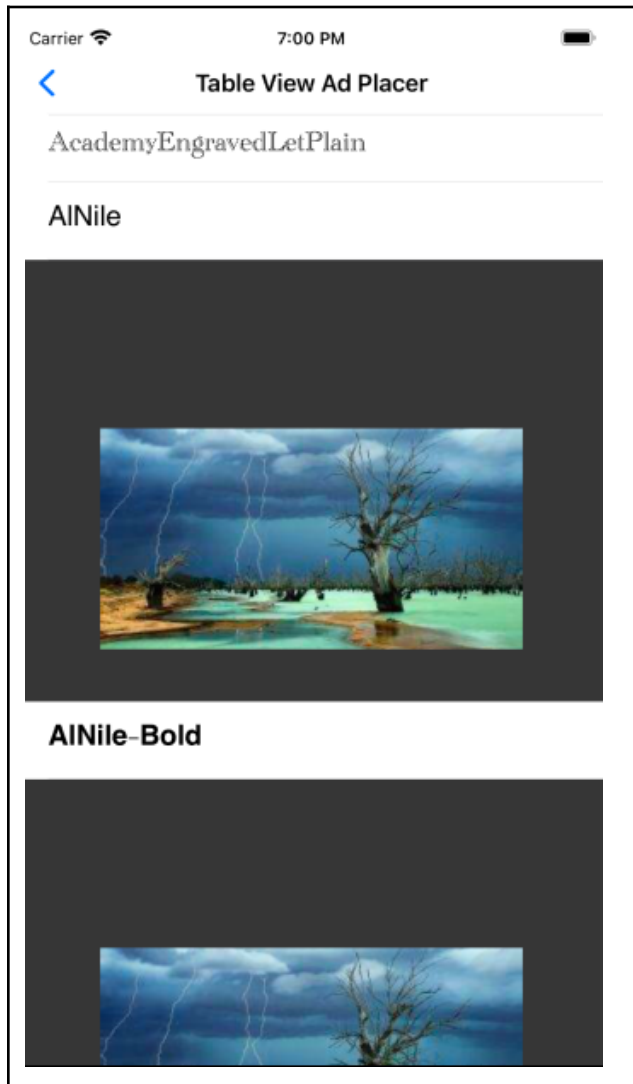
Interstitial Video Ad :

1. Select the interstitial video ad from the list and enter the zone id (**Ref Fig:1**), the interstitial video playing in full screen.



Native Image Ad :

1. Select the native tableview image ad from the list and enter the zone id (**Ref Fig:1**), the ad will playing in tableview screen



2. Geofence ad

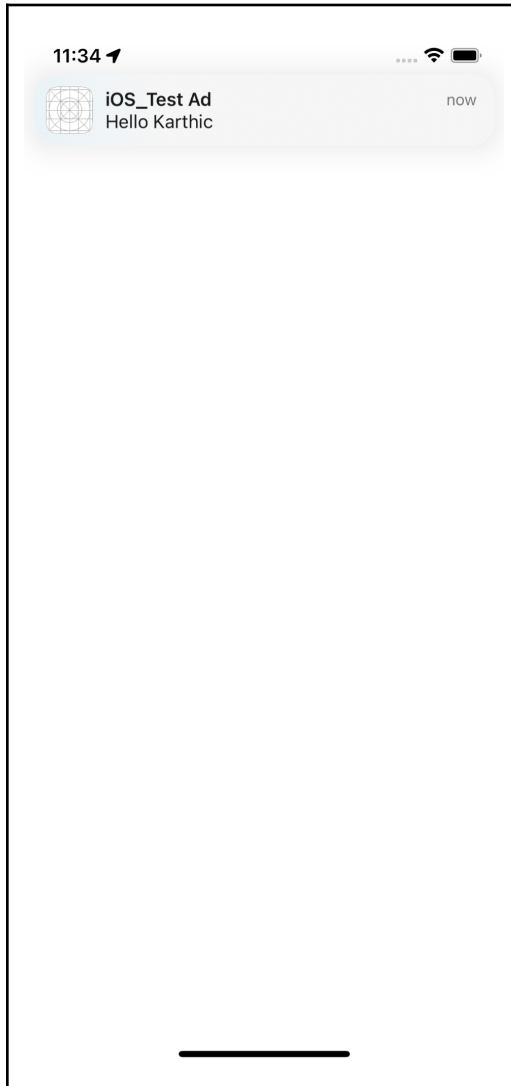
Geofence Interstitial Image Ad :

1. Select the Interstitial image ad from the list and enter the zone id (**Ref Fig:1**), the interstitial image will appear full screen.



Geofence Text Ad (Notification) :

2. Select the basic ad from the list and enter the zone id (**Ref Fig:1**), the notification will appear at the top of the screen.

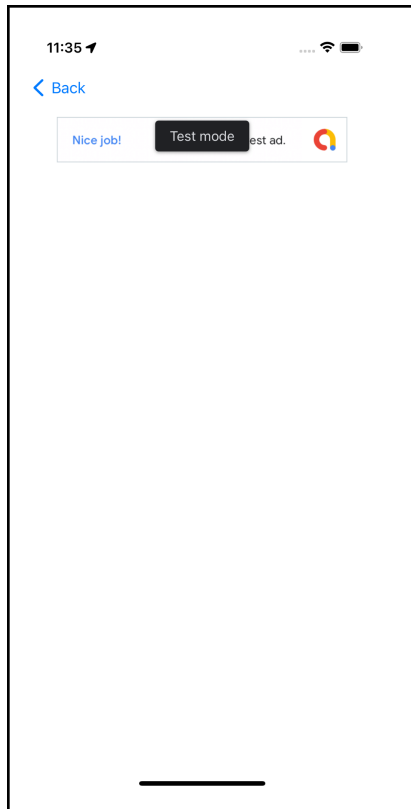


3. Mediation ad

Basics Ad :

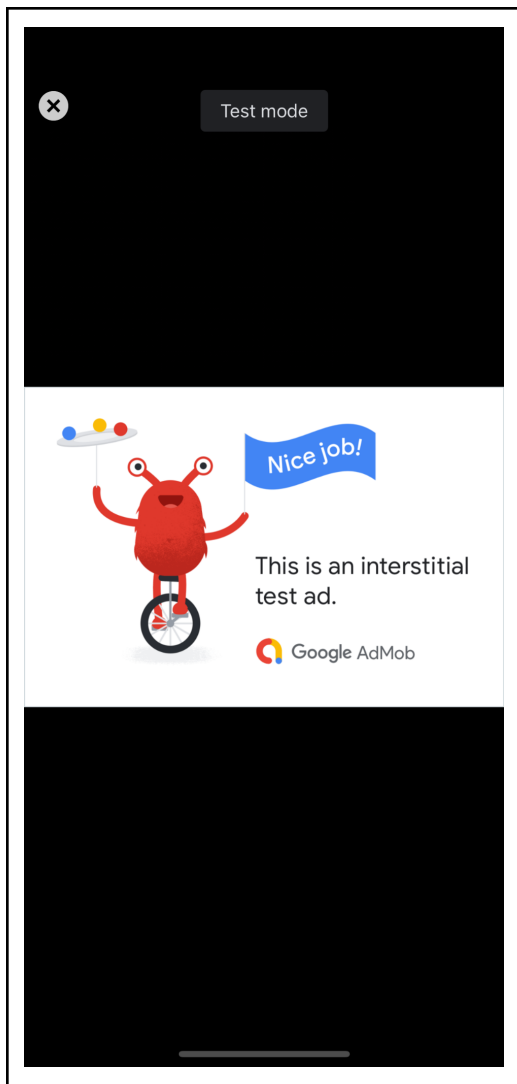
1. Select the Basic ad from the list and enter the zone id (**Ref Fig:1**), the banner will appear on top or bottom of the screen.

Basic ads include (Banner)



Interstitial Image Ad :

2. Select the Interstitial image ad from the list and enter the zone id (**Ref Fig:1**), the interstitial image will appear full screen.



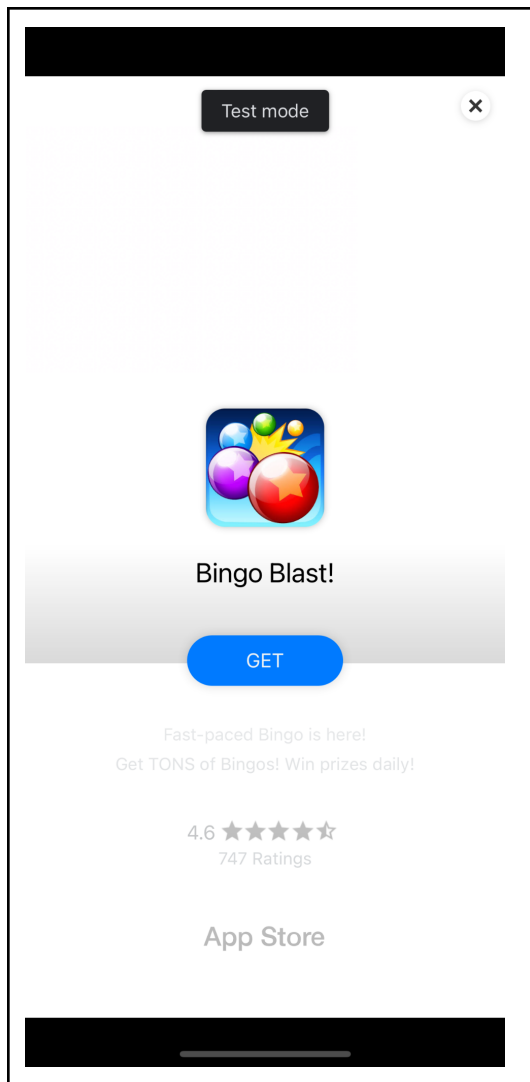
Rewarded Video Ad :

3. Select the rewarded video ad from the list and enter the zone id (**Ref Fig:1**), the rewarded video playing in full screen



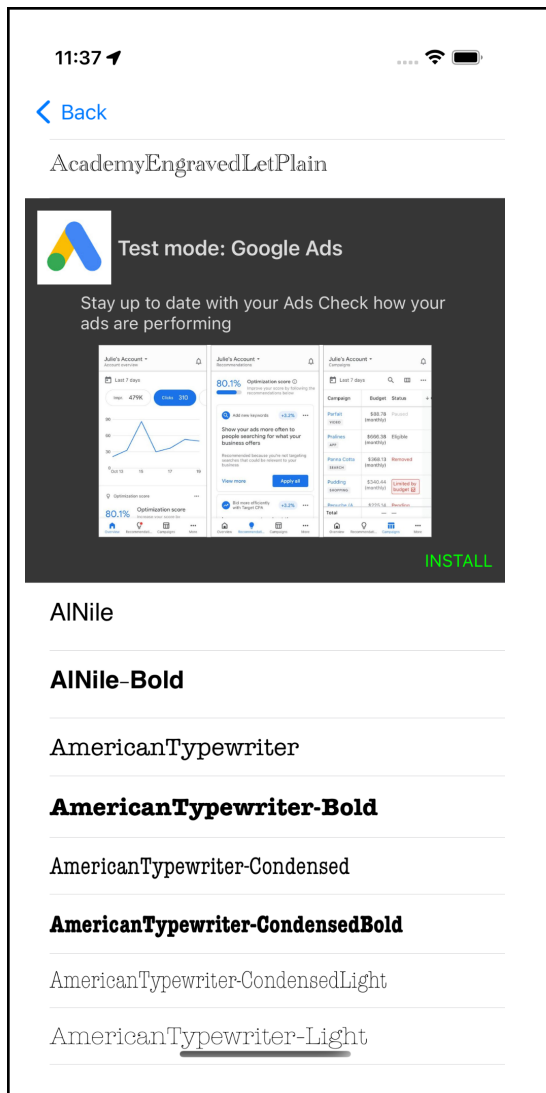
Interstitial Video Ad :

4. Select the interstitial video ad from the list and enter the zone id (**Ref Fig:1**), the interstitial video playing in full screen



Native Video Ad :

5. Select the native tableview image ad from the list and enter the zone id (**Ref Fig:1**), the ad will playing in tableview screen



----- END -----