

AdMob Adapter Installation

iOS Admob Adapter Installation

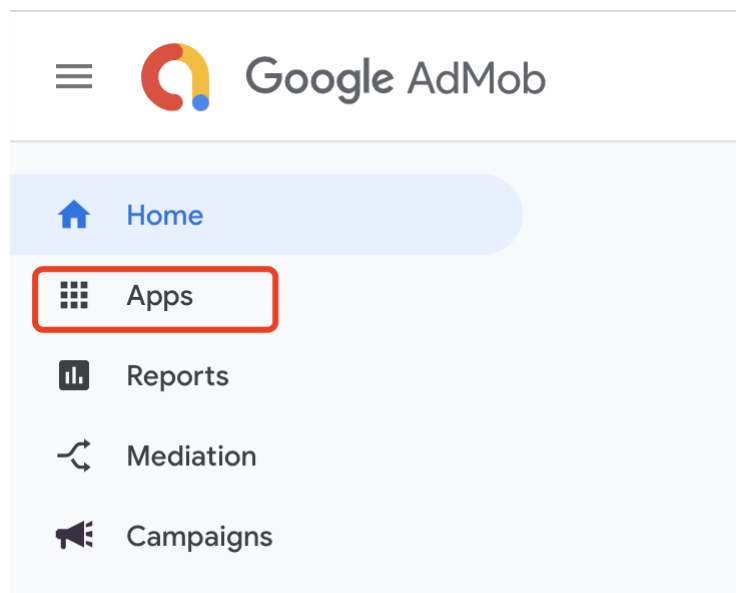
Adding AdView to AdMob

Step 1: Login to your AdMob account and configure AdView

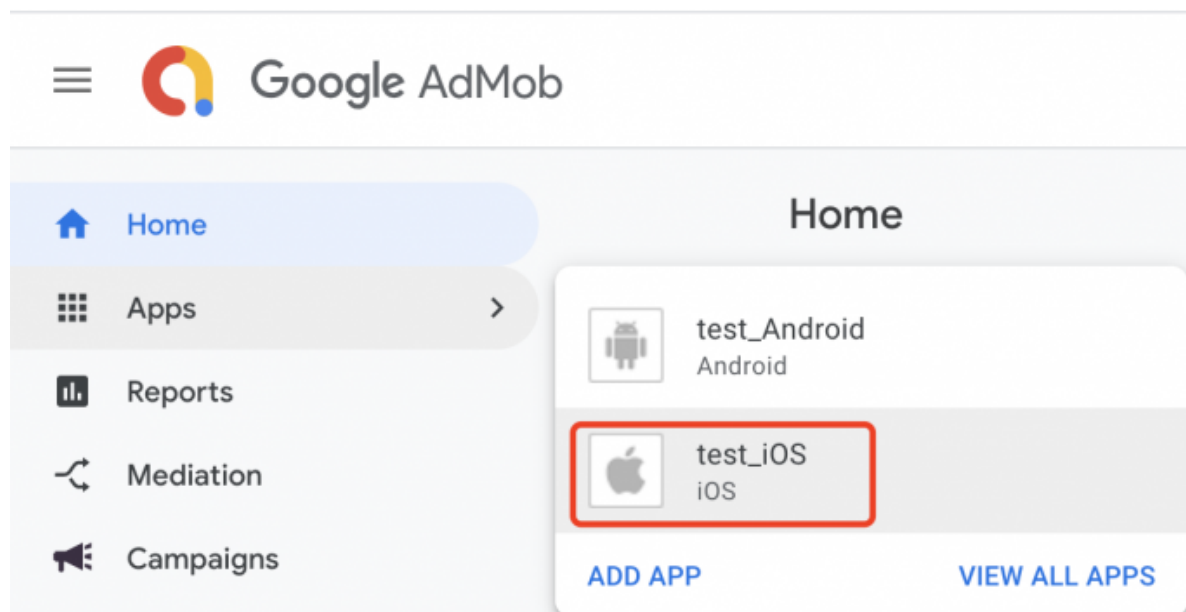
As explained above, you must have a AdMob account, and an associated app within this account, in order to integrate AdView.

Step 2: Select your app and click to monetize it

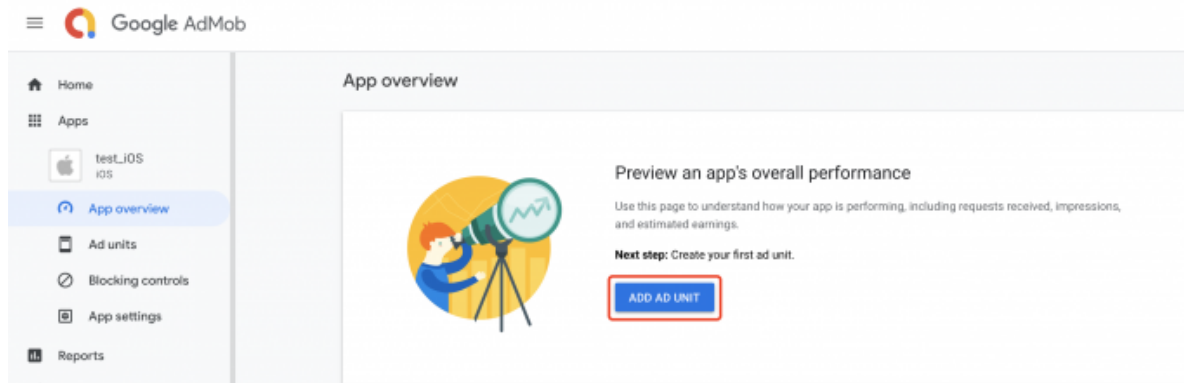
Within your AdMob account, press the 'Apps' tab



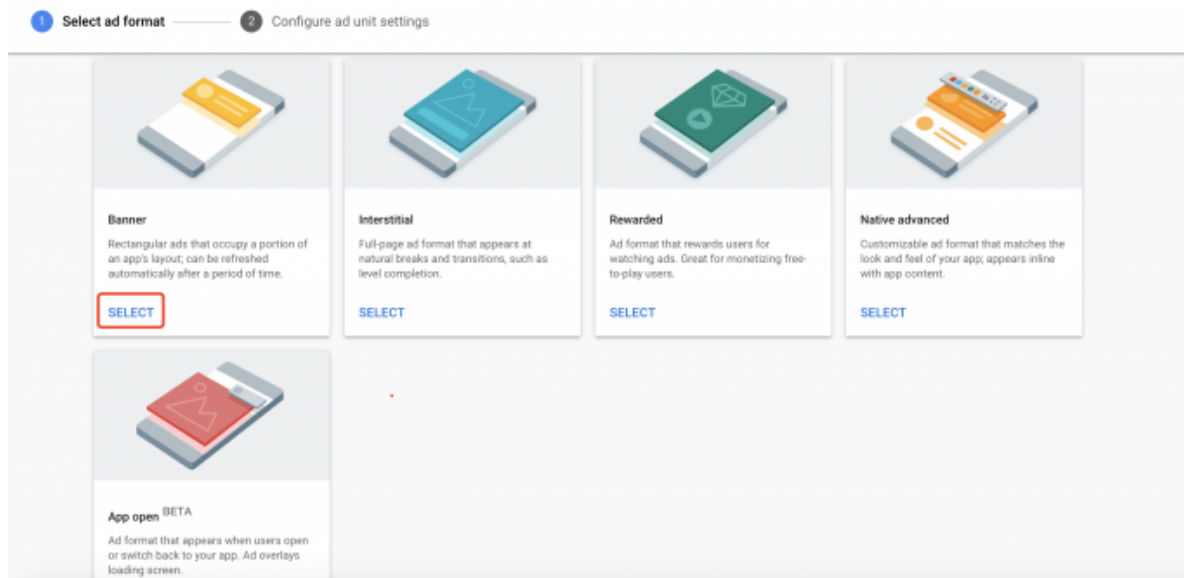
AdView must be integrated as a custom event. If your app does not have a ad unit yet, you should create a new ad unit by clicking on your app's name:



After clicking on your app's name, you will need to click the "New ad unit" button



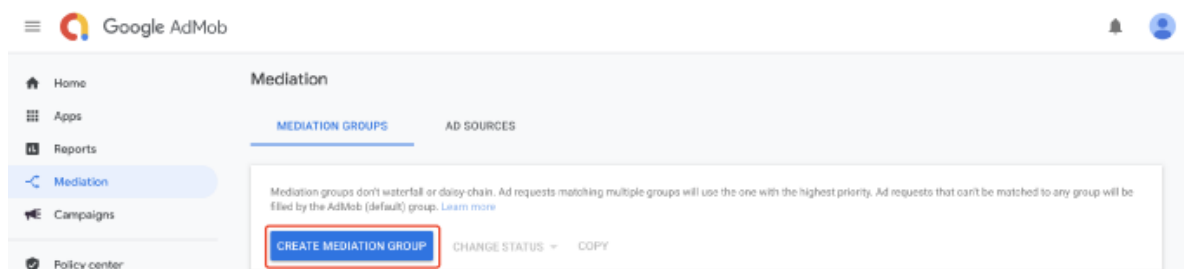
Provide a name for your new ad unit. After making the device type selection, you should choose a format that you need to installation:



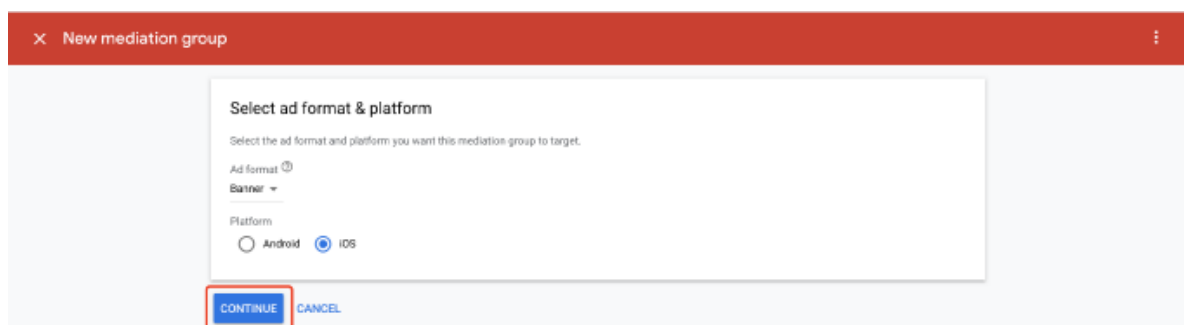
note: create MREC AD unit, you need select "Banner". The configuration is the same as Banner.

Step 3: Create Mediation group for your ad placement

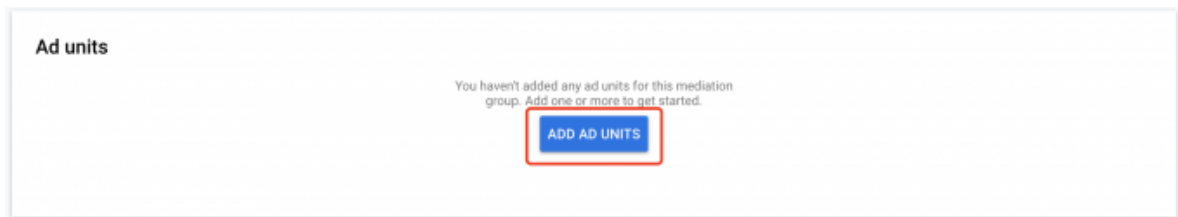
Within your AdMob account, press the 'Mediation' tab, and click "Create Mediation group".



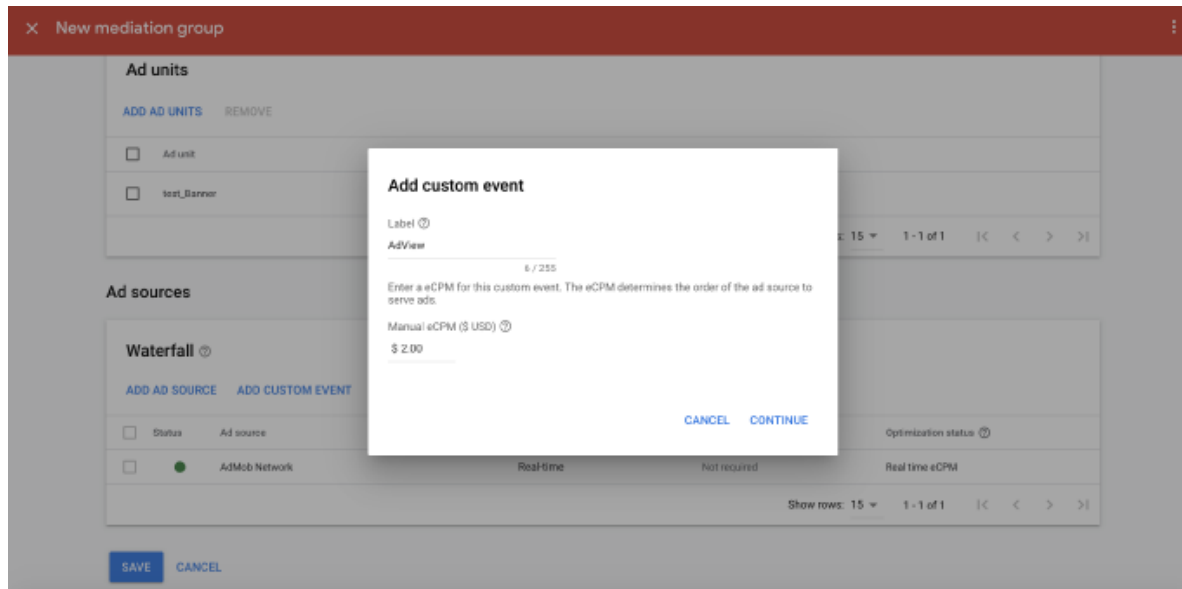
Then click "CONTINUE"



At this time, please provide a Mediation group name for your new Mediation group, then you need click"ADD AD UNITS"



AdView must be integrated as a custom event,so please click Add Custom Event.When you finish write Label and eCPM click continue



Add the following class to the "Class name" section for your ad unit.

Please note that for different types of ad unit, you need to set a different class,following are the classes corresponding to different ad unit types:

Map ad units: test_ios_banner_custom-event

Map your ad units to this custom event. ?

AdMob	test_ios_banner_custom-event
iOS test_ios_app iOS test_ios_banner_unit ca-app-pub-7209484307992421/64548...	Class Name AdViewCustomEventBanner Parameter (optional) SDK20191020vuy5lczp;POS...

CANCEL

DONE

Banner Class Name:

AdViewCustomEventBanner

MREC Class Name:

AdViewCustomEventBanner

Interstitial Class Name:

AdViewCustomEventInterstitial

Reward video Class Name:

AdViewCustomEventRewarded

Native Class Name:

AdViewCustomEventNative

It is mandatory to add your AdView appId (SDK-KEY) and placement ID (posId) to the "Parameter(optional)" section.

AdView appId (SDK-KEY);AdView placement ID (posId)

Now click 'DONE' and 'SAVE'

Ad sources

Waterfall ⓘ

[ADD AD SOURCE](#) [ADD CUSTOM EVENT](#) [CHANGE STATUS](#) ▼

☐	Status	Ad source	Order (by eCPM) ⓘ	Ad unit mapping ⓘ	Optimization status ⓘ
☐	●	AdMob Network	Real-time	Not required	Real time eCPM
☐	●	adview ✎	\$10.00 ✎	Edit	Not supported

Show rows: 15 ▼ 1 - 2 of 2 |< < > >|

[SAVE](#) [CANCEL](#)

Now you have completed the setup, and only need simple integration to display AdView ads.

AdMob Adapter Installation

iOS AdMob Adapter Installation

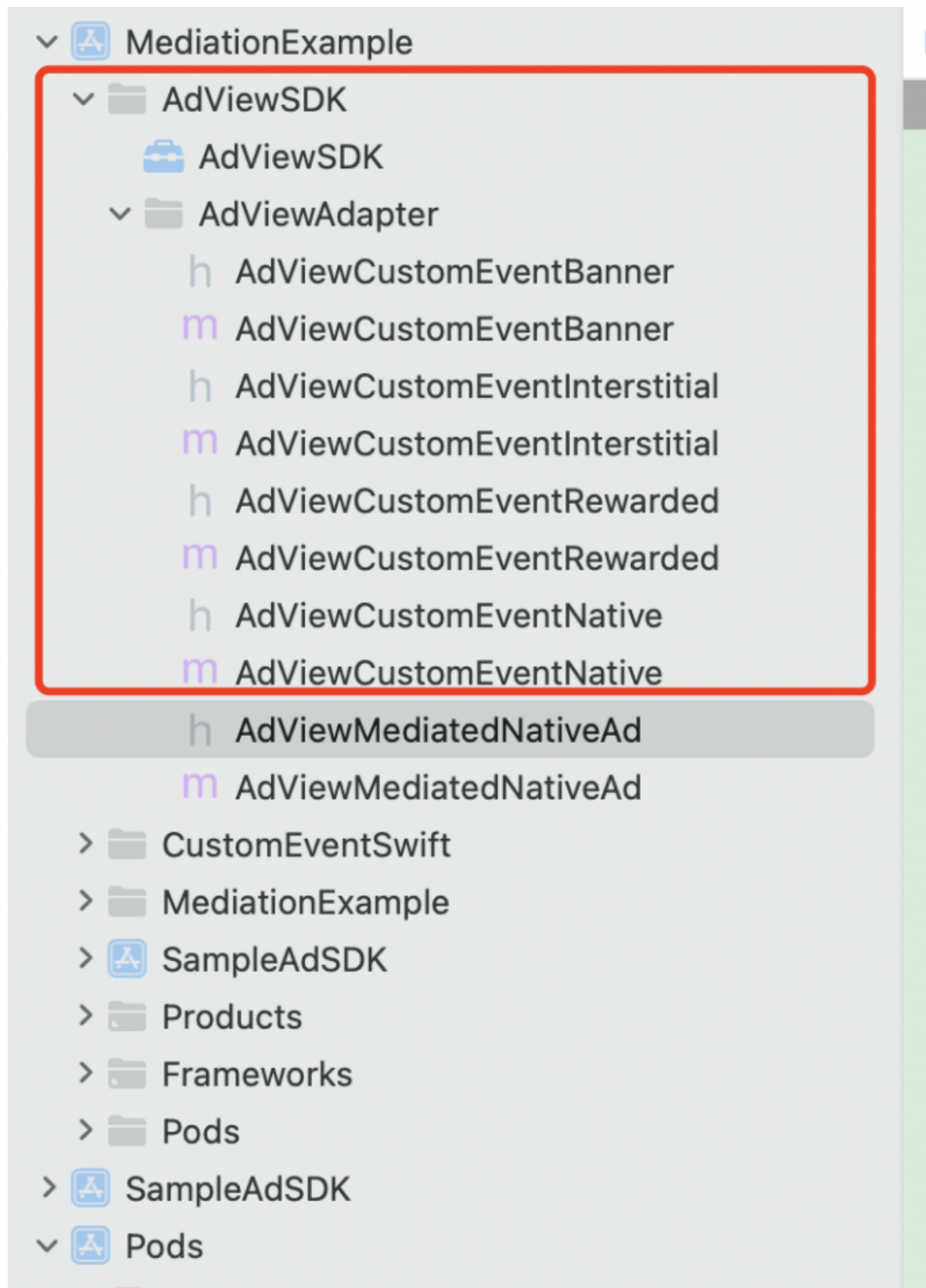
Step 1: Get the file with the adapter

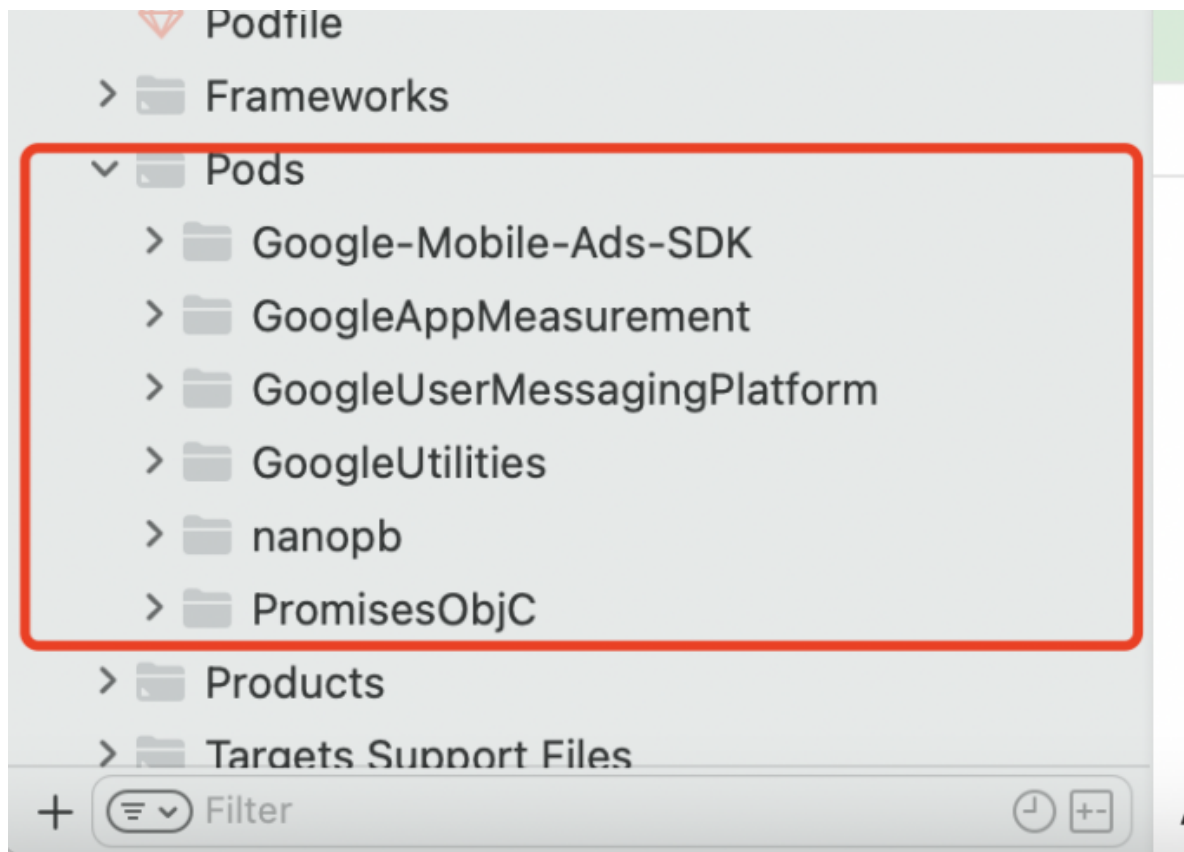
When you see this document, you should have obtained the following documents:

- AdViewAdapter file
- AdMob-AdViewAdapter-Example file

Otherwise please contact your AM or partner@adview.com.

Step 2: Add the AdView folder & AdViewSDK & AdMobSDK



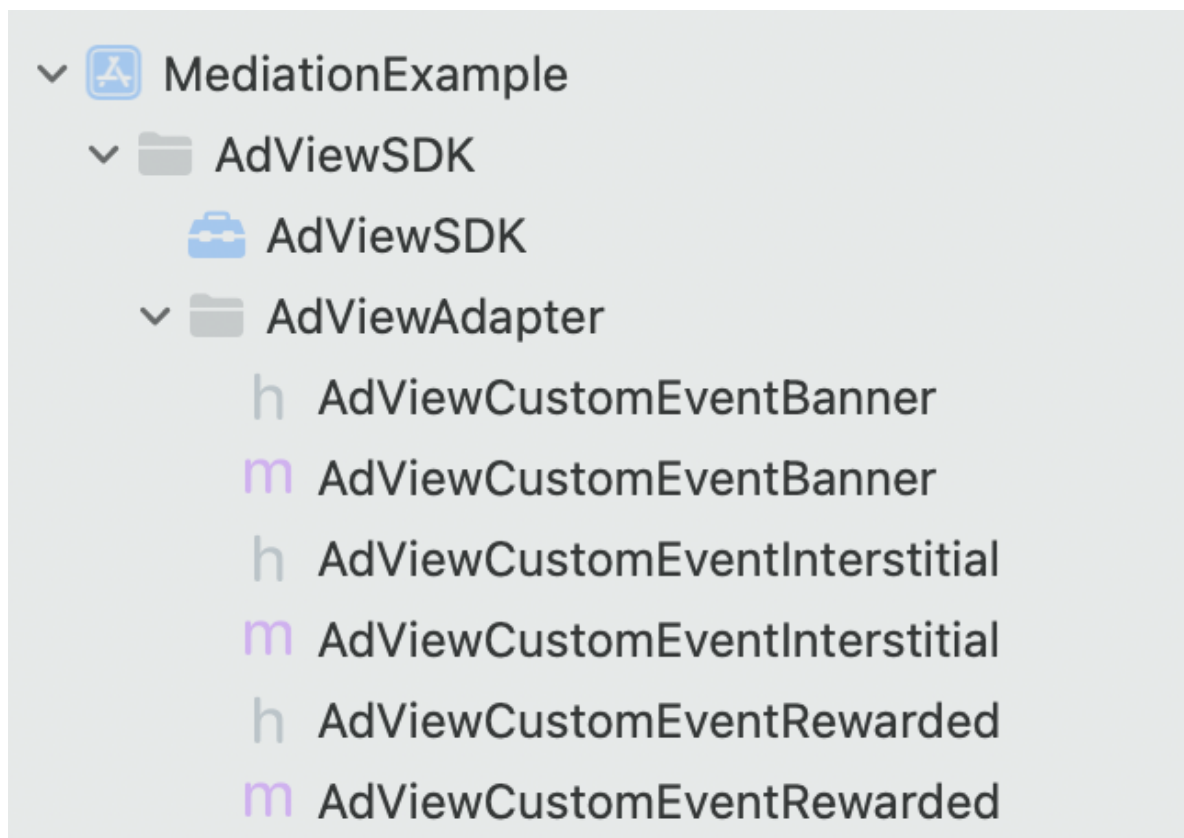


Step 3: Use AdMob sdk to display ADs

[AdMob quick-start](#)

Or refer to the demo:

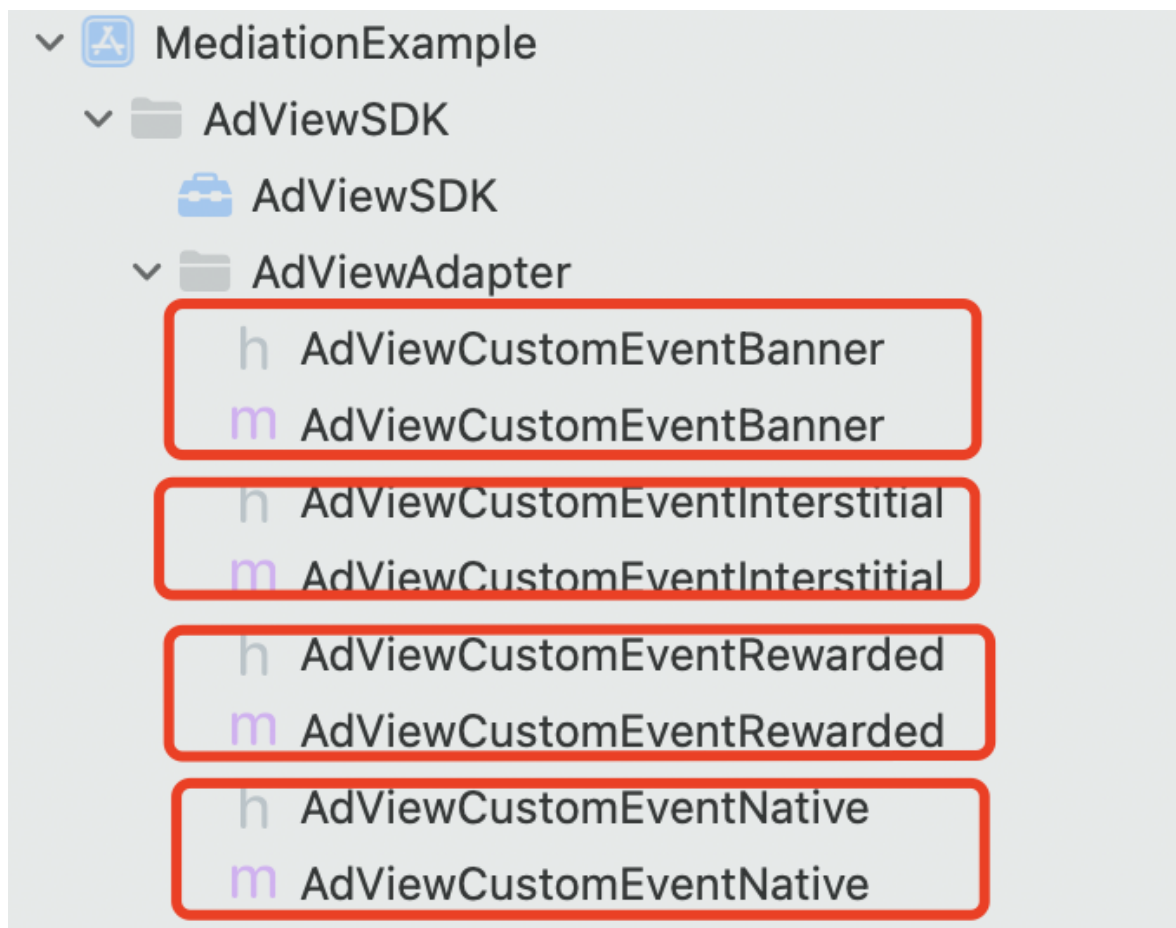
It include banner & mrec & interstitial & rewarded video & native ad ,
the AdSourceConfig file include AD unid and ViewController file include Implementation steps
,Open app Selection Objective-C Custom Event The AdView Ads are inspired



- h AdViewCustomEventNative
- m AdViewCustomEventNative
- h AdViewMediatedNativeAd
- m AdViewMediatedNativeAd
- >  CustomEventSwift
- ✓  MediationExample
 - h AppDelegate
 - m AppDelegate
 - h ExampleNativeAdView
 - m ExampleNativeAdView
 -  Images
 - h AdSourceConfig
 - m AdSourceConfig
 - h ViewController
 - m ViewController
- h StartViewController
- m StartViewController
-  Main
-  UnifiedNativeAdView
-  ExampleNativeAdView
-  LaunchScreen

Step 4: CustomEvent adapter module

At the AdView folder, all of the custom events at here:



Step 5: GDPR

AdViewSDK support GDPR protocol, you only need to implement its proxy method and pass in the corresponding value

```
@protocol AdViewViewDelegate <AdViewGDPRProcoto1V1, AdViewGDPRProcoto1V2>
@protocol AdViewVideoDelegate <AdViewGDPRProcoto1V1, AdViewGDPRProcoto1V2>
@protocol AdViewNativeAdDelegate <AdViewGDPRProcoto1V1, AdViewGDPRProcoto1V2>

@protocol AdViewGDPRProcoto1V1 <NSObject>
@optional
- (BOOL)CMPPresent;
- (BOOL)subjectToGDPR;
- (NSString *)userConsentString;
- (NSString *)parsedPurposeConsents;
- (NSString *)parsedVendorConsents;

@protocol AdViewGDPRProcoto1V2 <NSObject>
@optional
- (BOOL)GDPRApplies;
- (NSString *)TCFString;
-
```

Step 6: You're all done!

Your AdMob SDK should start showing AdView ads immediately.

Otherwise please contact your AM or partner@adview.com.