



Android Adapter Integration Guide

Overview/Updates

The AdColony Android AdMob/DFP Adapter can be used to integrate AdColony video ads into the Google Mobile Ads mediation platform. The following instructions assume you have already integrated the Google Mobile Ads SDK. If you have not done so, please refer to the [documentation](#).

- Server-side parameters are now supported.
- Calls to configure, pause, and resume are no longer required.
- We now support rewarded (V4VC) ad units via RewardedVideoAds.
- Both our V4VC and Interstitial ad units are directly supported via Interstitial requests.
- Use of our Instant-Feed ad units is possible via Banner requests and is explained below.
- You will also have full access to our [entire API](#) by including our [library](#) in your project.

Setup

First, download our library (with adapter built in) [here](#). Afterwards, follow the project setup instructions as explained [here](#).

Next, edit the mediation for your desired ad unit on the AdMob dashboard. Add AdColony as an ad source and set your app/zone ids. This should look something like the image below:

Edit AdColony

App ID Optional	<input type="text" value="app185a7e71e1714831a49ec7"/>
Client Options Optional	<input type="text"/>
Zone IDs (semicolon separated list) Optional	<input type="text" value="vz8d02e12ced0044a6b7;vze5b807162a"/>
Ad network optimization ?	<input type="checkbox"/> Automatically pull the eCPM value from this ad network
<input type="button" value="Continue"/>	<input type="button" value="Cancel"/>

Interstitials

Send along an extras Bundle with your request with AdColony's adapter class. Optionally, you can also set the zone id for the request using AdColonyBundleBuilder.

```
AdColonyBundleBuilder.setZoneId(ZONE_ID);

AdRequest adRequest = new AdRequest.Builder()
    .addNetworkExtrasBundle(AdColonyAdapter.class, AdColonyBundleBuilder.build())
    .build();
```

If you do not send along a Bundle with your zone id, we will instead use the first available zone included in your AdColony.configure() call.

Banners

- Build a banner ad request with an Instant-Feed zone id passed via AdColonyBundleBuilder.
 - If you simply add your AdView to the layout in onAdLoaded(), your placement will consist only of our video container.
 - For retrieving Instant-Feed metadata for a truly native experience, in your onAdLoaded() callback, check the if the ad is provided by AdColony:
 - if (ad.getMediationAdapterClassName().equals("com.jirbo.adcolony.AdColonyAdapter"))...
 - Treat this callback as a sign that an Instant-Feed ad is available and create/place an AdColonyNativeAdView object as explained in our documentation [here](#).
 - A demo app showing a simple integration can be found [here](#).
 - Please note that Instant-Feed ads will only be served to ad requests with the following AdSizes:
 - MEDIUM_RECTANGLE
 - Custom AdSize with width >= 300, height >= 200, and an aspect ratio between 4:3 and 16:9
-

Rewarded Video

- Build an AdRequest identical to the interstitial sample above.
- Use your rewarded video ad instance to load the ad request via the load() method.
- In the onRewardedVideoAdLoaded() callback, you can now show the ad using the show() method on your rewarded video ad instance.
- A demo app showing a simple integration can be found [here](#).

Notes

- AdColony is only supported on API 14 (ICS) and above, and only on devices with a memory class (app specific memory limit) of 32MB or greater.
- If you have any questions or issues with your integration, please email us at support@adcolony.com.