



Flurry Advertising

Flurry Android Adapter for MoPub

Adapter version 5.0.0.r1

Updated: 12/09/2014

Mediate Flurry through MoPub

To integrate Flurry as the Custom Native Network in the MoPub ad serving flow, you need the Custom Event Class code incorporated into your application in addition to the Flurry SDK. Three quick steps are necessary:

1. Integrate the Flurry SDK and Flurry adapter for MoPub code into your app
2. Configure Flurry's Ad space(s)
3. Configure MoPub to mediate Flurry

1. Integrate Flurry SDK and custom code into your app

1. If your application is not yet tracked by Flurry, create a new application on Flurry's dev portal. After logging into <https://dev.flurry.com>, select the Applications tab and from the top right-hand corner select Add New Application. In case your application is already tracked by Flurry, you can download the latest SDK from the adjacent top right-hand link.



2. Download the [Flurry Android SDK](#). Record the API Key found on the download page. This will identify your app in the Flurry system.

3. Add the Google Play Services SDK to your project. This is required for Android Advertising ID support. See <http://developer.android.com/google/play-services/setup.html>.

4. Add the jar files from the Flurry SDK (FlurryAndroidAnalytics-5.x.x.jar and FlurryAndroidAds-5.x.x.jar) to your project. Configure the build path of the project to include the jar files.

5. Add the Flurry MoPub adapter classes (found in the com.mopub.mobileads package) to your project.

6. If you plan to run [ProGuard](#) on your APK before releasing your app, you will need to add the following to your “proguard.cfg” file:

```
# Preserve Flurry
-keep class com.flurry.** { *; }
-dontwarn com.flurry.**
-keepattributes *Annotation*,EnclosingMethod
-keepclasseswithmembers class * {
public <init>(android.content.Context, android.util.AttributeSet, int);
}

# Preserve Flurry mediation classes for MoPub
-keep public class com.mopub.mobileads.FlurryCustomEventBanner
-keep public class com.mopub.mobileads.FlurryCustomEventInterstitial

# Google Play Services library
-keep class * extends java.util.ListResourceBundle {
    protected Object[][] getContents();
}

-keep public class
com.google.android.gms.common.internal.safeparcel.SafeParcelable {
    public static final *** NULL;
}

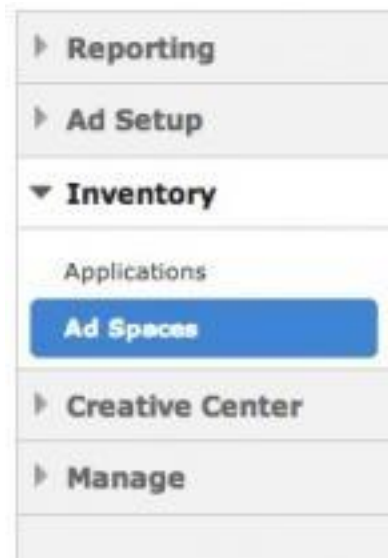
-keepnames @com.google.android.gms.common.annotation.KeepName class *
-keepclassmembernames class * {
    @com.google.android.gms.common.annotation.KeepName *;
}

-keepnames class * implements android.os.Parcelable {
    public static final ** CREATOR;
}
```

2. Configure Flurry Ad space(s)

For each MoPub ad unit that you would like to mediate Flurry through, please create a matching ad space on Flurry's dev portal (<http://dev.flurry.com>).

Log into the developer portal and navigate to the **Publishers** tab. On the left-hand navigation bar select **Inventory** and then **Ad Spaces**.



With Ad Spaces selected you'll see an index of previously created ad spaces. To set up a new one, Click on the New Ad Space button on the top right. The Ad Space setup screen has 4 modules.

The Basic Setup section includes fields required to define the name, application, dimensions, placement and orientation of the ad space.

The basic setup is all you need to start, you can click Save.

Please note that mediating Flurry through MoPub requires no additional Flurry related coding. The Flurry Advertising code is already incorporated in the com.mopub.mobileads package (added to your project in the previous step).

If you are integrating banner ads, and would like to streamline reconciliation of the impressions count, we recommend you turn off banner refresh on the Flurry side and let MoPub refresh the ads. ***This can be done by setting the refresh rate for the ad space to an unusually high number, like 9999 (instead of the default 30 seconds). This setting is found 4 section***

of the ad space setup - Advanced Options. Change the value for 'Ad Refresh Rate' setting.



The screenshot shows the 'Advanced Options' dialog box. The 'Ad Refresh Rate' field is highlighted with a red box. The field is labeled 'Ad Refresh Rate' and has a value of '3000' with a unit of '(seconds)'. Other settings visible include 'Maturity Rating', 'App Name', 'App Store Country', 'Advertiser Name', 'App Category', 'Price', 'Advertiser Domain', 'SDK Category', and 'Tag/Module'.

3. Configure MoPub to mediate Flurry

Flurry's custom events are implemented in accordance with instructions provided by MoPub (<https://github.com/mopub/mopub-android-sdk/wiki/Custom-Events>).

After you incorporate the Flurry files into your project (as described in the Step 1 of this guide), you need to configure Flurry as the Custom Native Network into your mediation flow. Please follow instructions provided by MoPub:

<http://help.mopub.com/customer/portal/articles/988568-setting-up-a-custom-native-network-campaign>.

Please note that the Flurry Custom Event classes are `com.mopub.mobileads.FlurryCustomEventBanner` for banners and `com.mopub.mobileads.FlurryCustomerEventInterstitial` for interstitial ads. You must also provide the Flurry API key (as created in the Step 1 of this guide) and ad space (as created in the Step 2 of this guide) in the Data (Options) section of the Custom Event for each ad unit you are mediating. The parameters are named "apiKey" and "adSpaceName". (for example, {"apiKey" : "<your api key here>", "adSpaceName": "MoPubBanner"}).