



Intégration des SDK Publicitaires pour formats IAB & Natif avec DoubleClick For Publishers (Android)

V1 – 26/10/2015

Contact LA PLACE MEDIA :

Jean-Christophe WACK (jcwack@laplacemedia.com - 06 60 57 79 31)

Team Mobile LPM (mobile@laplacemedia.com)

SOMMAIRE

1- VUE D'ENSEMBLE

2- INTEGRATION DE L'ADAPTER DFP → MOPUB

3- INTEGRATION DU SDK MOPUB

4- INTEGRATION DE L'ADAPTER MOPUB → ADNETWORK

- Adapter MoPub → InMobi
- Adapter MoPub → Facebook
- Adapter MoPub → Flurry
- Adapter MoPub → Google AdMob

5- INTEGRATION DES SDKS ADNETWORK

- SDK InMobi
- SDK Facebook
- SDK Flurry
- SDK Google AdMob

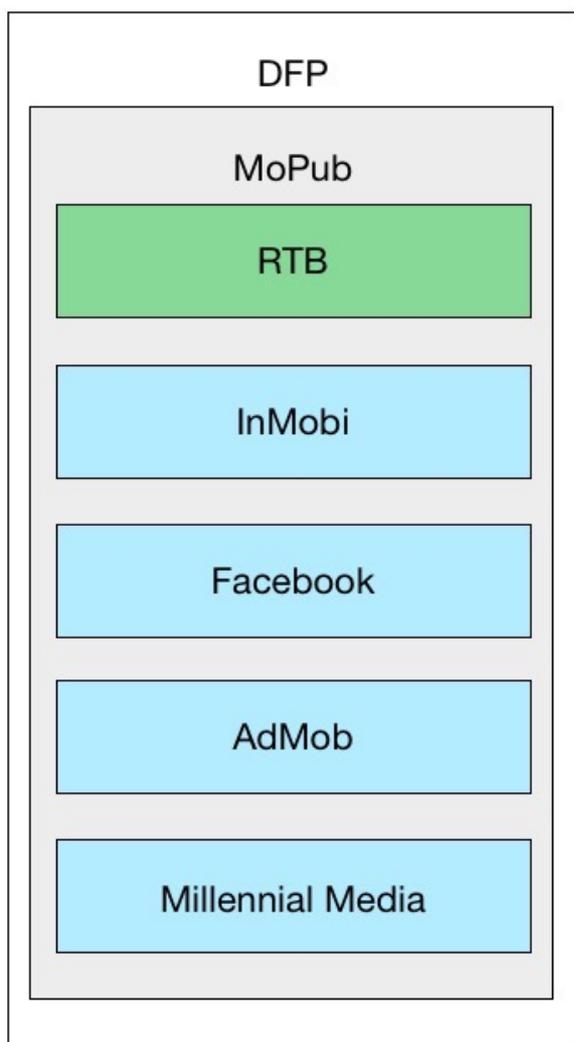
6- TEST D'INTEGRATION

1 VUE D'ENSEMBLE

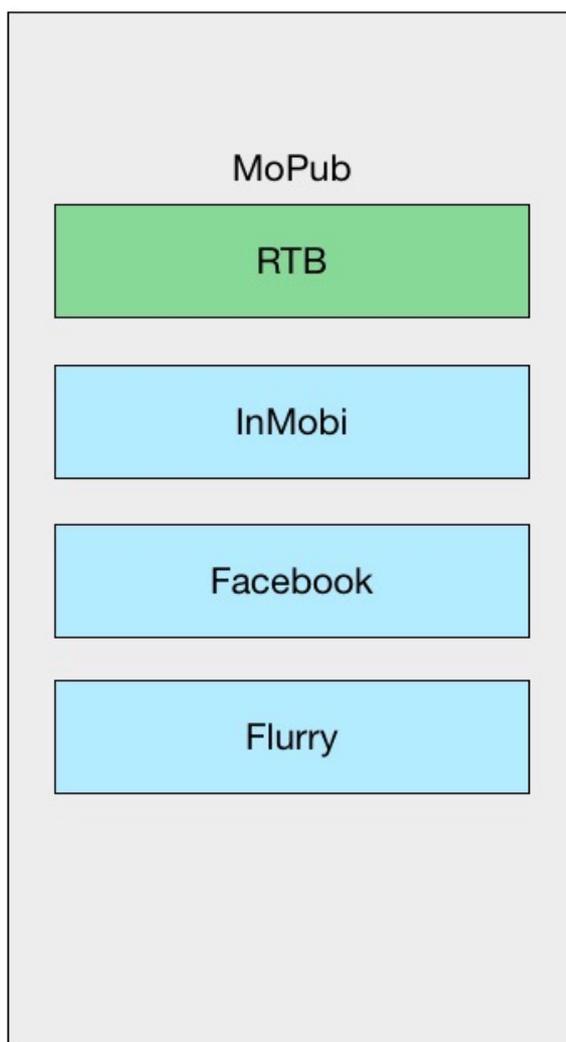
Ce document décrit l'intégration des SDK publicitaires La Place Media sous Android pour l'affichage de publicités aux formats IAB et Natifs (in-feed) lorsque l'AdServer primaire est DFP.

Une fois la procédure complétée, la situation sera la suivante :

- Sur les formats IAB (bannière et interstitiel), DFP sera l'AdServer primaire. MoPub sera l'AdServer secondaire et **unique** outil de médiation pour les AdNetworks suivants : Google AdMob, Millennial Media, InMobi et Facebook.
- Sur le format natif, MoPub sera l'AdServer primaire sur ce format et le **seul** outil de médiation au sein duquel seront intégrés les AdNetworks suivants : InMobi, Facebook et Flurry by Yahoo.



Intégration des formats IAB



Intégration du format natif

Attention : Pour les formats IAB, DFP propose d'intégrer différents acteurs en tant que « médiation network » au même niveau que MoPub. Seul MoPub doit être intégré comme « médiation network » au sein de DFP. Les autres acteurs seront yieldés dans MoPub.

Prérequis : Avant de vous lancer dans la procédure suivante, assurez-vous d'être en possession...

- du Placement ID MoPub nécessaires pour votre application

- du SDK Flurry correspondant au Placement ID MoPub évoqués ci-dessus
- du mock-up (fichier image) et du fichier .xml associé, correspondant à la mise en forme des créas natives
- du sample code créé par La Place Media
- d'un SDK DFP installé et fonctionnel dans votre application

2 INTEGRATION DE L'ADAPTER DFP → MOPUB

Téléchargez l'adapter DFP vers MoPub à l'adresse suivante :

http://mopub-dfp.mopub.com/android/AdMobMediation_Adapter_MoPub_Android_SDK.zip

La documentation pour intégrer cet adapter au sein de Xcode est disponible à l'adresse ci-dessous (points 2 et 3) :

<https://developers.google.com/mobile-ads-sdk/docs/dfp/android/mediation>

3 INTEGRATION DU SDK MOPUB

Attention : le SDK MoPub existe sous 2 formes (MoPub Full SDK et MoPub base SDK). Vous devez déployer le MoPub Full SDK (lien ci-dessous)

<http://s3.amazonaws.com/mopub-android-sdk/mopub-android-full-sdk.zip>

Le déploiement du SDK est décrit au lien ci-dessous (seules les instructions de la partie 'Incorporate the MoPub Android SDK' doivent être suivies).

<https://github.com/mopub/mopub-android-sdk/wiki/Getting-Started>

'Gradle Integration' & 'Maven Integration'

Ces 2 parties sont à suivre uniquement si vous utilisez l'un de ces outils.

'Adding Google Play Services'

Cette Rubrique doit être suivie rigoureusement. Le package 'API Level: Android 2.3' est nécessaire pour faire fonctionner la librairie.

'Adding Ad Network Libraries to Your Project'

Ne pas suivre cette partie (elle sera effectuée au point 5 'Intégration des SDKs AdNetwork')

'Updating your Android Manifest'

Suivre la documentation pour le Full Bundle. L'ajout du dernier tag a normalement **déjà été effectué dans la partie 'Adding Google Play Services'**

```
<meta-data android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
```

'Using the GitHub Repository Source Code'

Ne pas suivre cette rubrique.

'Displaying Ads'

Ne pas suivre cette partie pour les formats IAB (bannière et interstitiel), les appels MoPub étant directement effectués par DFP + Adapter. Il faudra par contre suivre les instructions équivalentes pour le format natif (« Integrating Native Ads » : <https://github.com/mopub/mopub-android-sdk/wiki/Native-Ads-Integration>).

Finalement, les propriétés de votre projet devraient au moins inclure les librairies suivantes :

type filter text

- ▶ Resource
 - Android**
 - Android Lint Preferences
 - Builders
 - Java Build Path
- ▶ Java Code Style
- ▶ Java Compiler
- ▶ Java Editor
- Javadoc Location
- Project References
- Run/Debug Settings
- Task Repository
- Task Tags
- XML Syntax

Android

Project Build Target

Target Name	Vendor	Platform	API ...
<input type="checkbox"/> Android 2.3.1	Android Open Source Project	2.3.1	9
<input type="checkbox"/> Google APIs	Google Inc.	2.3.1	9
<input type="checkbox"/> Android 4.4.2	Android Open Source Project	4.4.2	19
<input type="checkbox"/> Android 4.4W.2	Android Open Source Project	4.4W.2	20
<input checked="" type="checkbox"/> Android 5.0.1	Android Open Source Project	5.0.1	21

Library

Is Library

Reference	Project
✓ ..\mopub-sdk	mopub-sdk
✓ ..\google-play-services_lib	google-play-services_lib

Buttons: Add..., Remove, Up, Down, Restore Defaults, Apply, OK, Cancel

4 INTEGRATION DE L'ADAPTER MOPUB → ADNETWORK

Télécharger le zip suivant :

<https://github.com/mopub/mopub-android-sdk/archive/master.zip>

Pour les formats IAB, Les adaptors InMobi se trouvent dans le dossier suivant :

/extras/src/com/mopub/simpleadsdemo

Les adaptors des autres adnetworks se trouvent dans le dossier suivant :

/extras/src/com/mopub/mobileads

Pour le format natif, les adaptors InMobi et Facebook se trouvent dans le dossier suivant :

/extras/src/com/mopub/nativeads

INTEGRATION DE L'ADAPTER MOPUB → INMOBI:

Pour les formats IAB, récupérer les fichiers : InMobiBanner.java et InMobiInterstitial.java et les déployer dans le package *mopub-sdk/src/main/java/com.mopub.mobileads*

Dans les fichiers, il vous faut ensuite renseigner en dur dans le code les App ID InMobi que nous vous transmettons, dans la ligne *String inMobiAppId = "YOUR_INMOBI_APP_ID";*

Pensez aussi à renommer le package name en haut de ces deux fichiers en *package com.mopub.mobileads;*

Pour le format natif, déposez le fichier InMobiNative.java sous *mopub-sdk/src/main/java/com/mopub/nativeads* (package name: *com.mopub.nativeads*)

INTEGRATION DE L'ADAPTER MOPUB → FACEBOOK:

Pour les formats IAB, Récupérer les fichiers : FacebookBanner.java et FacebookInterstitial.java et les déployer dans le package *mopub-sdk/src/main/java/com.mopub.mobileads*

Pour tester la bonne intégration de Facebook et voir apparaitre les test ads, il vous faudra intégrer la ligne *AdSettings.addTestDevice("xxx"); après placementId = serverExtras.get(PLACEMENT_ID_KEY);*

Où xxx est l'identifiant de votre device. Celui-ci sera disponible dans la console LogCat lors de l'appel des ads Facebook.

Pour le format natif, déposez le fichier FacebookNative.java dans votre projet sous *mopub-sdk/src/main/java/com/mopub/nativeads* (package name: *com.mopub.nativeads*)

INTEGRATION DE L'ADAPTER MOPUB → FLURRY:

Au lien <https://github.com/flurry/FlurryAdapterForMoPubAndroid>

Les adaptors Flurry se trouvent dans le dossier suivant:

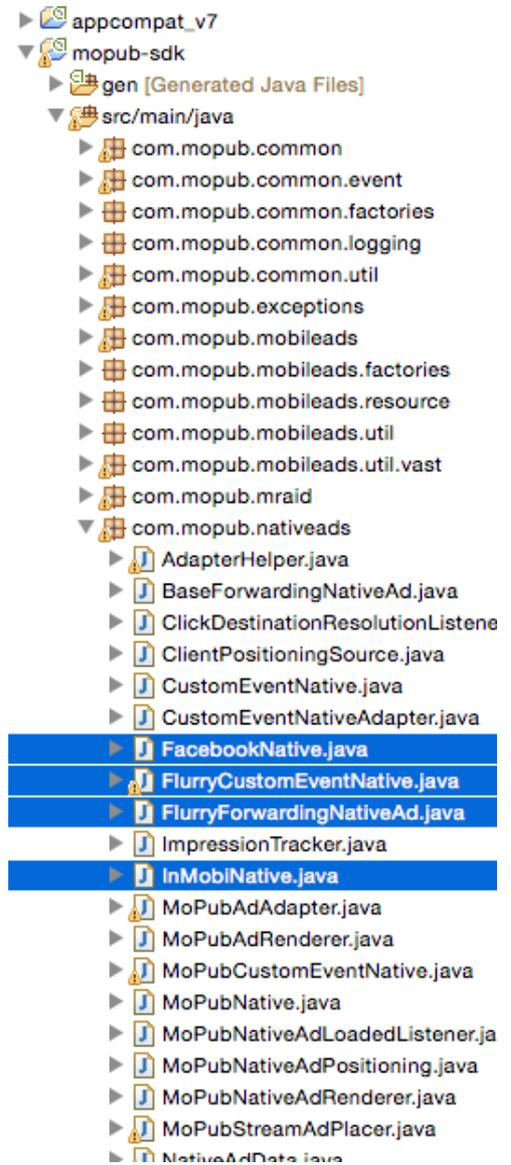
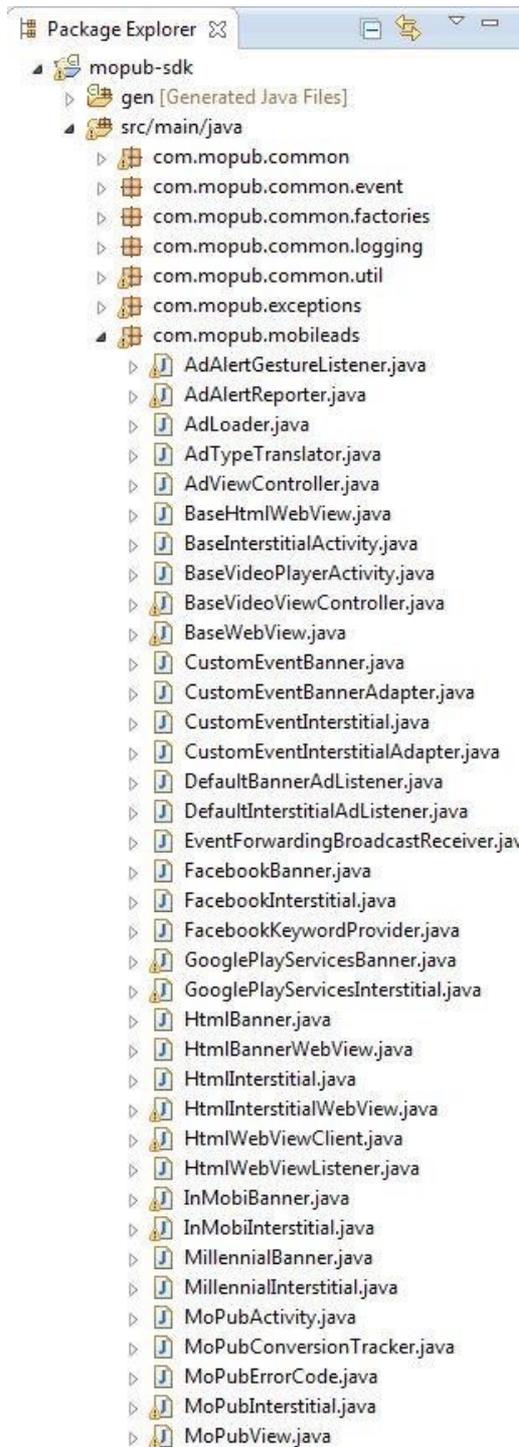
/src/com/mopub/nativeads

Déposer les fichiers FlurryCustomEventNative.java et FlurryForwardingNativeAd.java dans votre projet sous `mopub-sdk/src/main/java/com/mopub/nativeads` (package name: `com.mopub.nativeads`)

INTEGRATION DE L'ADAPTER MOPUB → GOOGLE ADMOB:

Récupérer les fichiers : `GooglePlayServicesBanner.java` et `GooglePlayServicesInterstitial.java` et les déployer dans votre package `mopub-sdk/src/main/java/com.mopub.mobileads`

Vous devriez donc avoir dans l'arborescence d'Eclipse la configuration suivante (IAB à gauche puis Natif à droite) :



5 INTEGRATION DES SDKS ADNETWORK

INTEGRATION DU SDK INMOBI:

Veillez télécharger le SDK InMobi au lien ci-dessous :

https://dl.inmobi.com/SDK/InMobi_Android_SDK.zip

Importez le fichier .jar que vous trouverez dans le dossier /InMobi_Android_SDK_4.5.3 dans votre projet. Nous vous recommandons de placer le SDK InMobi dans le dossier suivant :

mopub-sdk/libs

Veillez suivre **uniquement** les instructions de la rubrique 'Configuration' de la procédure ci-dessous

<https://www.inmobi.com/support/integration/23817448/22051163/android-sdk-integration-guide/>

LA PLACE MEDIA a déjà paramétré le compte InMobi (création compte et placement IDs).

Dans la partie 'Manifest File Changes', seule la sous partie 'Mandatory Activities' est absolument indispensable. Les autres parties sont soit optionnelles, soit elles ont déjà été effectuées au moment du déploiement du SDK MoPub.

Il vous faudra aussi inclure le SDK InMobi comme librairie du projet mopub-sdk, en faisant click droit sur le projet mopub-sdk > Java Build Path > Libraries > Add JARs... > Browse jusqu'à mopub-sdk/libs/InMobi-4.5.3.jar

Note : Si vous avez déjà intégré le SDK InMobi lors de l'intégration des formats bannière et/ou interstitiel, il faudra bien sûr ne pas refaire cette étape.

INTEGRATION DU SDK FACEBOOK:

Veillez télécharger le SDK Facebook au lien ci-dessous :

<https://developers.facebook.com/docs/android>

Dézippez le fichier sur votre ordinateur, puis importez le fichier 'AudienceNetwork.jar' se trouvant sous /facebook-android-sdk-3.23.0/AudienceNetwork/bin dans le dossier suivant :

mopub-sdk/libs

Il vous faudra ensuite inclure le SDK Facebook comme librairie du projet mopub-sdk, en faisant click droit sur le projet mopub-sdk > Java Build Path > Libraries > Add JARs... > Browse jusqu'à mopub-sdk/libs/AudienceNetwork.jar

Suivez la **Step 2** disponible à cette adresse pour la suite de l'intégration :

<https://developers.facebook.com/docs/audience-network/getting-started>

LA PLACE MEDIA a déjà paramétré le compte Facebook (création compte et placement IDs).

Note : Si vous avez déjà intégré le SDK Facebook lors de l'intégration des formats bannière et/ou interstitiel, il faudra bien sûr ne pas refaire cette étape.

INTEGRATION DU SDK FLURRY:

Veillez nous demander le SDK Flurry spécifique à votre application.

Importez les fichiers 'FlurryAds-X.Y.Z.jar' et 'FlurryAnalytics-X.Y.Z.jar' dans le dossier suivant :
mopub-sdk/libs

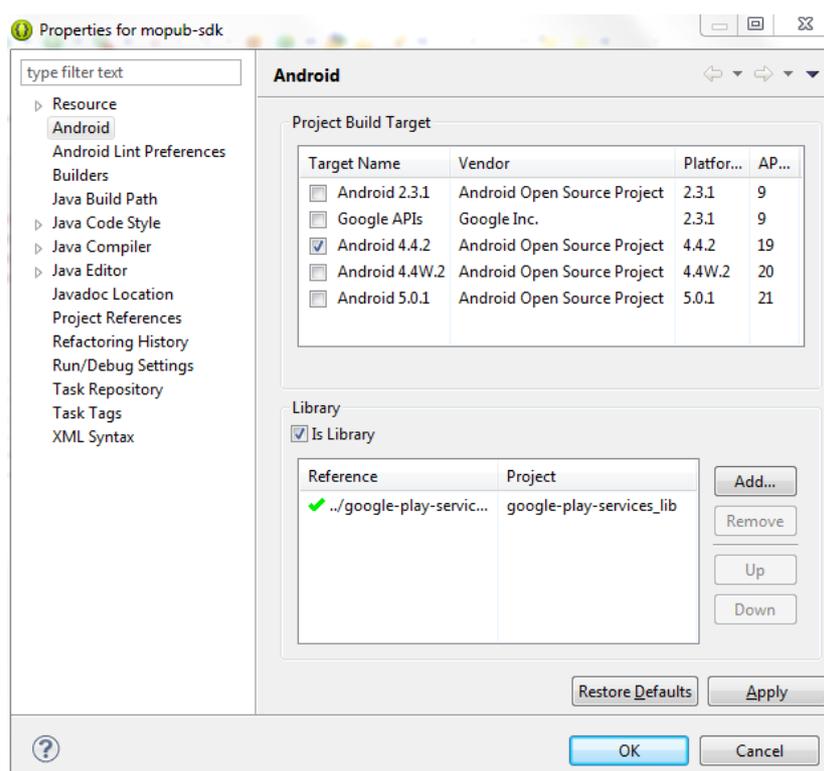
Il vous faudra ensuite inclure le SDK Flurry (les 2 fichiers) comme librairie du projet mopub-sdk, en faisant click droit sur le projet mopub-sdk > Java Build Path > Libraries > Add JARs... > Browse jusqu'à mopub-sdk/libs/FlurryAds-X.Y.Z.jar et mopub-sdk/libs/FlurryAnalytics-X.Y.Z.jar

Suivez les points 10, 11 et 12 (si vous travaillez avec proguard) de la partie 'Integrate Flurry SDK and Custom Code Into Your App' disponible au lien suivant pour modifier votre Manifest File :
<https://developer.yahoo.com/flurry/docs/publisher/gettingstarted/mediation/mopubmediatesflurry/android/>

La Place Media a déjà paramétré le compte Flurry (création compte et placement IDs).

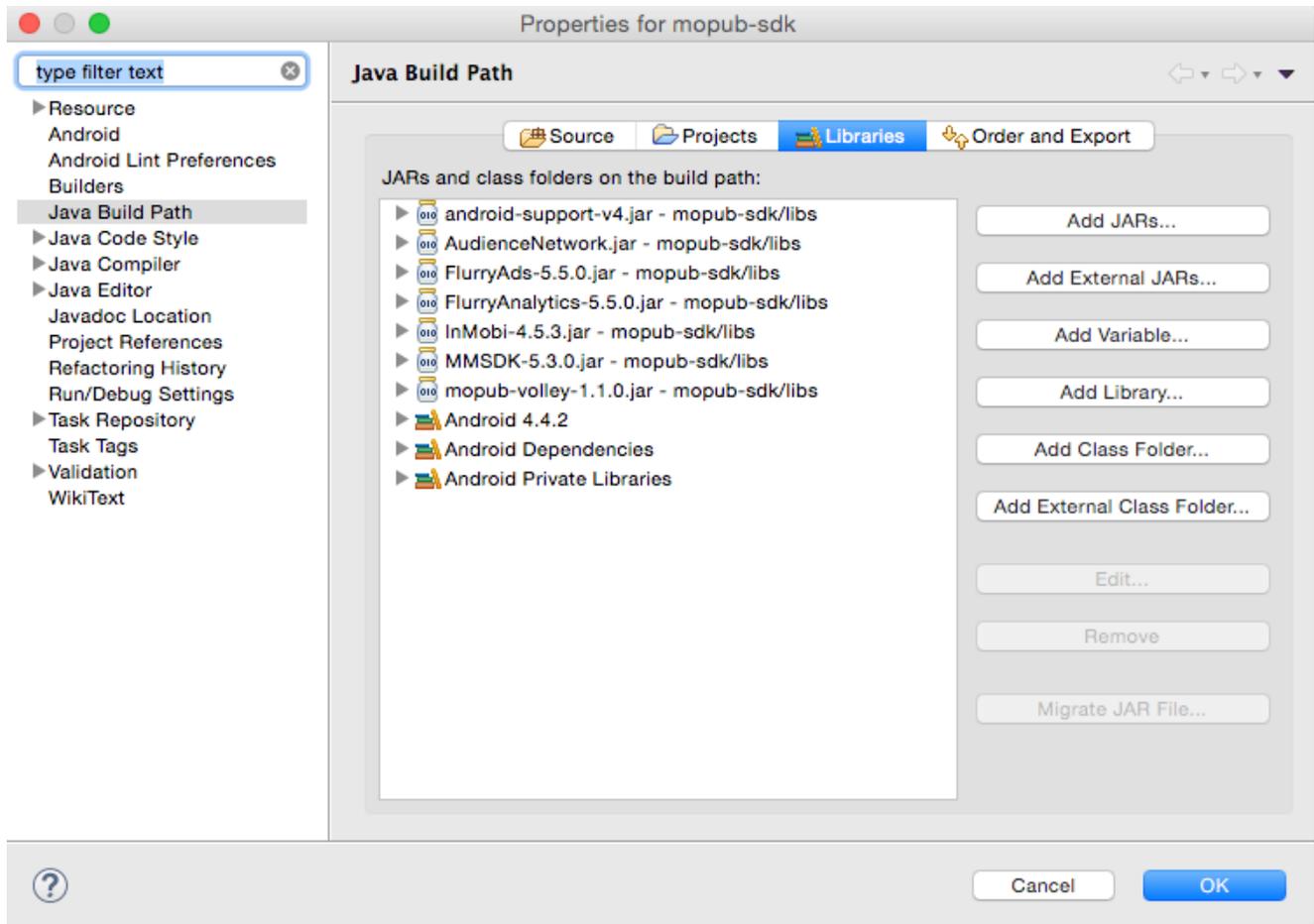
INTEGRATION DU SDK GOOGLE ADMOB:

Le fait d'avoir importé Google Play Services vous permet dès à présent de diffuser via Google AdMob, simplement grâce aux adaptateurs. Il vous suffit d'ajouter Google Play Services comme librairie Android dans les propriétés du projet mopub-sdk (en plus de votre projet courant où il devrait déjà être).



Un mot sur **Millennial Media** : Le SDK ainsi que les adaptateurs sont déjà présents dans le SDK MoPub (version Full) et il n'y a donc rien à faire dans le code de l'application.

Finalement, les propriétés du projet mopub-sdk devraient ressembler à cela :



Et le manifest de votre application devrait au moins intégrer les lignes présentes dans celui-ci :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.lpm.mynativeapp"
    android:versionCode="1"
    android:versionName="1.0" >

<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-sdk
    android:minSdkVersion="16"
    android:targetSdkVersion="22" />

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <activity
        android:name=".MainActivity"
        android:label="@string/app_name" >
        <intent-filter android:label="@string/app_name">
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".DisplayActivity"
        android:label="@string/title_activity_display_message"
        android:parentActivityName=".MainActivity" >
        <meta-data
            android:name="android.support.PARENT_ACTIVITY"
            android:value=".MainActivity" />
    </activity>
    <activity
        android:name=".SecondActivity"
        android:label="@string/second_activity_label" >
    </activity>

    <activity android:name="com.mopub.mobileads.MoPubActivity" android:configChanges="keyboardHidden|orientation|screenSize"/>
    <activity android:name="com.mopub.mobileads.MraidActivity" android:configChanges="keyboardHidden|orientation|screenSize"/>
    <activity android:name="com.mopub.common.MoPubBrowser" android:configChanges="keyboardHidden|orientation|screenSize"/>
    <activity android:name="com.mopub.mobileads.MraidVideoPlayerActivity" android:configChanges="keyboardHidden|orientation|screenSize"/>
    <activity android:name="com.google.android.gms.ads.AdActivity" android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|smallestScreenSize" />
    <activity android:name="com.millennialmedia.android.MMAActivity" android:theme="@android:style/Theme.Translucent.NoTitleBar"
    android:configChanges="keyboardHidden|orientation|keyboard" />
    <activity android:name="com.millennialmedia.android.VideoPlayer" android:configChanges="keyboardHidden|orientation|keyboard" />

    <activity android:name="com.inmobi.androidsdk.IMBrowserActivity"
        android:configChanges="keyboardHidden|orientation|keyboard|smallestScreenSize|screenSize"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"
        android:hardwareAccelerated="true" />

    <activity
        android:name="com.flurry.android.FlurryFullscreenTakeoverActivity"
        android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenSize|smallestScreenSize">
    </activity>

</application>

</manifest>
```

6 TEST D'INTEGRATION

Assurez vous d'avoir bien renseigné votre ID MoPub à l'endroit indiqué dans le document.

Enfin, une fois que votre application est prête à être testée, envoyez-nous un lien pour télécharger votre App de test (.apk) et convenons d'un rendez-vous téléphonique pour effectuer les tests ensemble.

En cas d'incompréhension au cours de la lecture de ce document ou de n'importe laquelle des étapes de la procédure, **n'hésitez surtout pas à nous contacter**.

Jean-Christophe WACK (jcwack@laplacemedia.com - 06 60 57 79 31)

Team Mobile LPM (mobile@laplacemedia.com)