

# **AMoAd SDK 3.0.11 Guide for Android**

2015 年 4 月 28 日

## 更新履歴

更新日	バージョン	更新内容
2014.01.28	3.0.0	コードレス、安定化、パフォーマンスアップ
2014.02.19	3.0.1	WeakReference バグ対応、リソースの解析バグ対応
2014.02.27	3.0.2	HTML 広告のスクロールを非活性化
2014.06.11	3.0.3	Google Mediation 対応
2014.07.11	3.0.4	Proguard のバグ対応
2014.07.25	3.0.5	Android1.6~2.2 対応
2014.08.20	3.0.6	SID バリデーションチェック対応
2014.08.21	3.0.7	V3.0.5 のバグ対応
2015.02.27	3.0.8	クリックしても遷移ができない Html 広告の対応
2015.03.24	3.0.9	Iframe 内のリンクをクリックしても遷移ができない Html 広告の対応
2015.04.24	3.0.10	Iframe 内のリンクをクリックすると Iframe 内が遷移される不具合の対応
2015.04.28	3.0.11	CalledFromWrongThreadException バグ対応

# 目次

<b>概要</b>	<b>4</b>
<b>1. 対応環境</b>	<b>4</b>
<b>2. ファイル構成</b>	<b>4</b>
<b>組み込み方法</b>	<b>5</b>
プロジェクトの設定	5
Google Play services	5
AndroidManifest.xml	7
<b>実装方法</b>	<b>8</b>
<b>コールバック</b>	<b>9</b>
<b>ローテーションの開始、停止</b>	<b>10</b>
<b>3. テスト検証について</b>	<b>10</b>
<b>APPENDIX A. API仕様</b>	<b>11</b>
AMoAdView	11
AdCallback インターフェース	12
<b>APPENDIX B. 広告をソースコードで生成する</b>	<b>13</b>
サンプルコード	13

## 概要

AMoAd SDK for Android(以下、本 SDK と表記します)は、Android デベロッパーが自らのアプリケーションに AMoAd が配信する広告を表示させるためのツールキットです。AMoAd Web サイトで発行する固有 ID を指定してアプリケーションに個別の広告を配信します。

広告は Android デベロッパーが画面レイアウトで指定する任意の位置に表示することが可能です。

## 1. 対応環境

本 SDK は以下のバージョンの Android OS をサポートします。

OS バージョン	API Level	備考
2.3.3	10	
3.0	11	タブレット専用
3.1	12	タブレット専用
3.2	13	タブレット専用
4.0	14	
4.0.3	15	
4.1.2	16	
4.2.2	17	
4.3	18	
4.4	19	

## 2. ファイル構成

本 SDK は以下のファイルから構成されます。



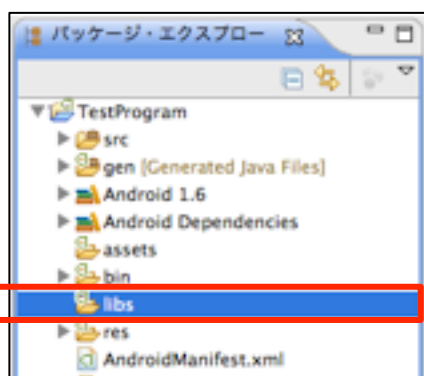
AMoAd\_3\_0\_11.jar ..... ライブラリ本体  
AMoAd\_SDK\_3\_0\_11\_Guide\_for\_Android.pdf 本ドキュメント  
AMoAdSample ..... ライブラリを使用したサンプルプログラム

## 組み込み方法

※ ADT のバージョンによりライブラリの組み込み方法が異なることがあります。ADT のバージョンに合った方法で組み込んでください。

### プロジェクトの設定

- ① libs フォルダを作成し、AMoAd\_3\_0\_11.jar を libs にドラッグ & ドロップします



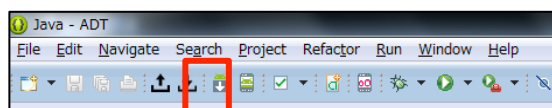
- ② AMoAd\_3\_0\_11.jar がライブラリに追加されます

※ 自動的に追加されない場合はプロジェクトのプロパティより手動で追加してください。

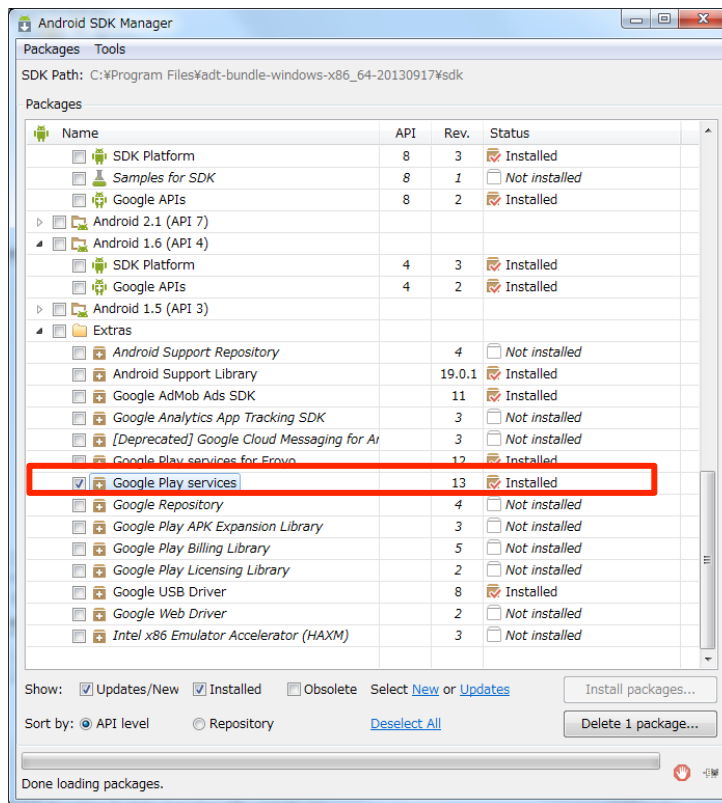
### Google Play services

本 SDK では追跡型広告を表示するため Google Advertising ID を使用します。Google Play services は Android SDK の Extras として Google より提供されています。以下に導入方法を示します。

- ③ Android SDK Manager を開く

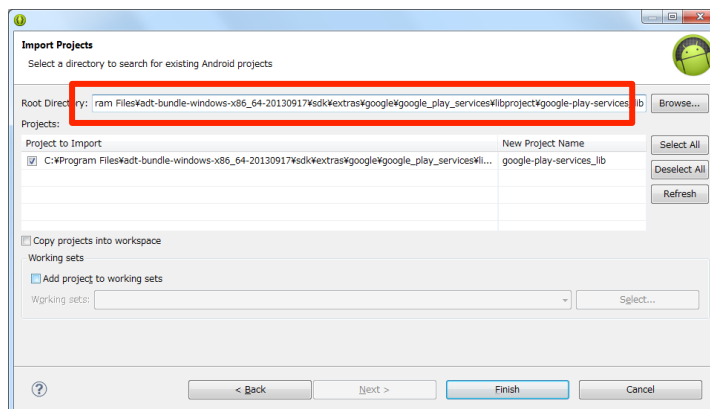


#### ④ Google Play services をインストールする



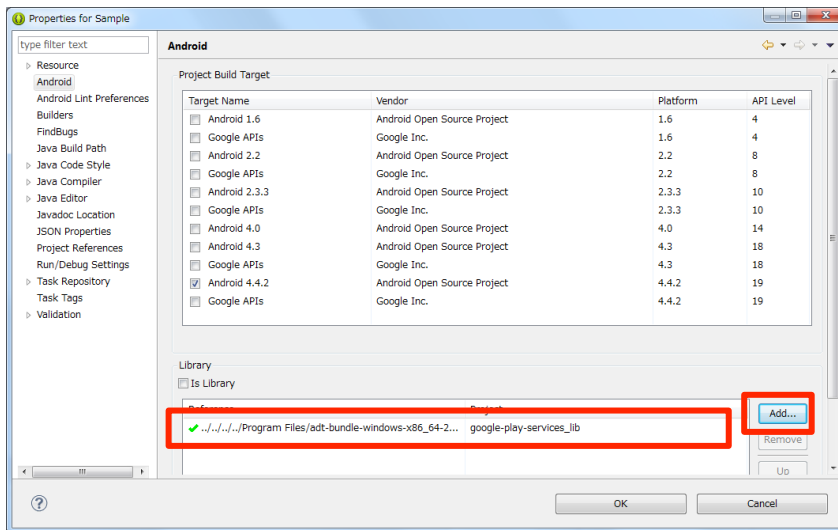
#### ⑤ Google Play services プロジェクトを Import する

メニュー「File > Import...」、「Android > Existing Android Code Into Workspace」  
Google Play services のパスを入力して「Finish」ボタンを押します。



## ⑥ Google Play services をライブラリに追加する

「Add...」ボタンから Import した Google Play services を選択します。



## AndroidManifest.xml

### ① メタ情報 (meta-data) を追加する

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

    <uses-sdk android:minSdkVersion="10" />

    <uses-permission android:name="android.permission.INTERNET" />

    <application android:allowBackup="true" android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" android:theme="@style/AppTheme">
        <meta-data android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />

        <activity android:name="com.example.amoadsample.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="com.example.amoadsample.SubActivity" />
    </application>

</manifest>
```

## 実装方法

本 SDK は layout.xml に AMoAdView タグを配置し属性を設定することで画面に広告を表示することができます。

layout.xml に指定する属性はソースコードから findViewById()などのメソッドで AMoAdView を取得し、セッターメソッドにより設定することもできます。詳細は APPENDIX A「API 仕様」を参照ください。

### ① レイアウト設定

```
layout.xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <com.amoad.AMoAdView
        android:id="@+id/banner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="#000"
        xmlns:amoad="http://schemas.amoad.com/apk/res/android"
        amoad:click_transition="jump"
        amoad:rotate_transition="rotate"
        amoad:sid="管理画面から発行される SID を設定してください" />

    <Button
        android:id="@+id/nextBtn"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="次へ" />

</LinearLayout>
```

### ② 項目説明

キー	値	
<b>xmlns:amoad</b>	"http://schemas.amoad.com/apk/res/android" ( <b>※必須</b> )	
<b>amoad:sid</b>	管理画面から発行される SID を設定してください ( <b>※必須</b> )	
<b>amoad:rotate_transition</b>	広告ローテーション時のトランジションを設定する	
	"none"	トランジション「なし」(デフォルト)
	"alpha"	トランジション「透過」
	"rotate"	トランジション「回転」
	"scale"	トランジション「拡大」
<b>amoad:click_transition</b>	広告クリック時のトランジションを設定する	
	"none"	トランジション「なし」(デフォルト)
	"jump"	トランジション「ジャンプ」
<b>android:background</b>	最初の広告情報が取れるまでに表示する画像を設定する	



## コールバック

AdCallback インターフェースを用いて SDK からの通知を受け取ることが可能です。  
通知の種類は以下の通りです。

メソッド	通知内容
didReceiveAd()	受信成功
didFailToReceiveAdWithError()	受信失敗
didReceiveEmptyAd()	空広告の受信

### ※ サンプルコード

```
AMoAdView amoadView = ((AMoAdView)findViewById(R.id.banner));

amoadView.setCallback(new AdCallback() {
    @Override
    public void didReceiveAd() {
    }
    @Override
    public void didFailToReceiveAdWithError() {
    }
    @Override
    public void didReceiveEmptyAd() {
    }
});
```

## ローテーションの開始、停止

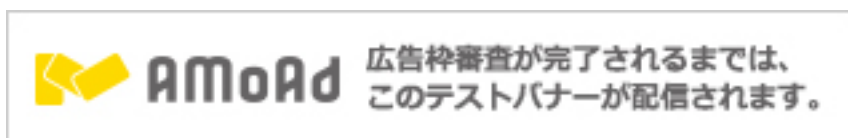
本 SDK は、管理画面の「自動更新間隔」で指定された間隔で自動的に広告をローテーション表示します。ローテーションさせたくない場合や AdMob 広告ネットワークメディエーション等の外部機能で AMoAd SDK を制御する際は、管理画面の「自動更新間隔」に「0」を設定してください。

※ 本 SDK でローテーションを止めたり間隔を制御することはできません。

## 3. テスト検証について

組み込みが正しく行われたかどうかは、管理画面から発行される sid を設定していただくことで、確認いただけます。

広告枠を作成いただくと、AMoAd で広告枠の審査を行わせていただきます。広告枠の審査が完了するまでは、以下のバナーが表示されます。



広告枠の審査が完了すると、広告主から提供されている広告が配信されるようになります。

## APPENDIX A. API 仕様

AMoAd SDK for Android で公開されている API を以下にまとめます。

※ AMoAdView をソースコードで生成する場合は、必ず Activity のコンテキストを指定してください。

### AMoAdView

<b>void setSid(String sid)</b>	
機能	AMoAd Web サイトで発行される ID(必須)を設定する
パラメータ	文字列 (String)

<b>void setRotateTransition(RotateTransition transition)</b>		
機能	ローテーション時のトランジションを設定する	
パラメータ	RotateTransition.NONE	トランジション「なし」
	RotateTransition.ALPHA	トランジション「透過」
	RotateTransition.ROTATE	トランジション「回転」
	RotateTransition.SCALE	トランジション「拡大」
	RotateTransition.TRANSLATE	トランジション「スライド・アップ」

<b>void setClickTransition(ClickTransition transition)</b>		
機能	クリック時のトランジションを設定する	
パラメータ	ClickTransition.NONE	トランジション「なし」
	ClickTransition.JUMP	トランジション「ジャンプ」

<b>void setCallback (AdCallback callback)</b>	
機能	通知を受け取るインターフェースを設定する。
パラメータ	AdCallback インターフェースを実装したクラスのインスタンス

以下のメソッドは対応する新しいメソッドに変更されました。以前のメソッドも使用することは可能ですが、非推奨となり、今後のバージョンでは使用できなくなる可能性があります。

変更前	型	変更後	型
setRotationAnimation	int	setRotateTransition	RotateTransition
setClickAnimation	boolean	setClickTransition	ClickTransition

以下のメソッドは非推奨となり呼び出しても何も起こりません。

requestFreshAd()、startRotation()、stopRotation()
---

## AdCallback インターフェース

<b>void didFailToReceiveAdWithError ()</b>	
機能	広告受信に失敗した場合に呼び出されます。
パラメータ	なし
戻り値	なし

<b>void didReceiveEmptyAd ()</b>	
機能	空広告を受信した場合に呼び出されます。
パラメータ	なし
戻り値	なし

<b>void didReceiveAd()</b>	
機能	広告受信に成功した場合に呼び出されます。
パラメータ	なし
戻り値	なし

## APPENDIX B. 広告をソースコードで生成する

本 SDK では広告の生成をソースコードで行うこともできます。その場合でもシングルトン・パターンなどで生成を管理する必要はなく、各画面に同じ固有 ID を設定することで広告を内部的に共有します。API の詳細については「APPENDIX A API 仕様」を参照ください。

### サンプルコード

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    AMoAdView amoadView = new AMoAdView(this);
    amoadView.setSid("管理画面から発行される SID を設定してください");
    amoadView.setRotateTransition(AMoAdView.RotateTransition.ALPHA);
    amoadView.setClickTransition(AMoAdView.ClickTransition.JUMP);

    ViewGroup content = (ViewGroup) findViewById(R.id.banner);
    content.addView(amoadView);
}
```

各画面で同じ固有 ID の広告を生成して追加します。