

# Android Banner v3.07开发者文档(zh)

## [1. SDK 嵌入步骤](#)

### [1.1 准备工作](#)

[Step 1.将 youmi-android.jar 导入您的工程中。](#)

[Step 2.在 manifest.xml 文件中配置用户权限](#)

[Step 3.在 AndroidManifest.xml 中添加 AdActivity](#)

[Step 4.初始化账号信息](#)

[Step 5.设置发布渠道号【可选】](#)

### [1.2 常用布局方式](#)

#### [1.2.1 使用 xml 布局嵌入广告](#)

[Step 1.在布局 xml 中设置广告条容器](#)

[Step 2.XML 布局代码部分](#)

#### [1.2.2 使用代码布局嵌入广告](#)

#### [1.2.3 使用悬浮布局嵌入广告](#)

## [2.常见问题](#)

### [2.1 版本更新内容](#)

### [2.2 广告条的尺寸定义](#)

### [2.3 有米广告 Android SDK 使用哪种字符编码](#)

### [2.4 有米广告 Android SDK 兼容 Android 系统 SDK 的哪些版本](#)

### [2.4 关于 AdViewListener 接口](#)

### [2.5 关于广告没有成功展示的问题排查](#)

### [2.6 关于混淆 APK 出错的问题](#)

### [2.7 更多常见问题](#)

## [3. 附录:API 文档](#)

### [3.1 net.youmi.android.AdManager](#)

### [3.2 net.youmi.android.AdView](#)

### [3.3 net.youmi.android.AdViewListener](#)

### [3.4 net.youmi.android.AdActivity](#)

# 1. SDK 嵌入步骤

## 1.1 准备工作

**Step 1.**将 **youmi-android.jar** 导入您的工程中。

- 1 右键您的工程根目录，选择□New□，再选择□Folder□，输入 Folder name 为□libs□
- 2 将 youmi-android.jar 拷贝到工程根目录的 libs 文件夹下
- 3 右键 youmi-android.jar ，选择□Build Path□再选择□Add to Build Path□
- 4 导入成功

**Step 2.**在 **manifest.xml** 文件中配置用户权限

请务必配置以下权限,否则将有可能获取不到广告。

- 7 android.permission.INTERNET,连接网络权限 INTERNET ， 用于请求广告
- 8 android.permission.READ\_PHONE\_STATE,用于精确统计用户手机的系统信息
- 9 android.ACCESS\_NETWORK\_STATE,用于精确识别网络接入点等信息
- 10 android.permission.ACCESS\_COARSE\_LOCATION,有助于精准投放地域广告以及帮助统计使用应用程序的用户的地区分布情况
- 11 android.permission.WRITE\_EXTERNAL\_STORAGE,有助于实现图片资源的缓存,节省流量,并可获得更好的用户体验
- 12 com.android.launcher.permission.INSTALL\_SHORTCUT,用于支持一些新的广告形式

请将下面权限配置代码复制到 AndroidManifest.xml 文件中：

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE">
</uses-permission>
<uses-permission android:name="android.permission.READ_PHONE_STATE">
</uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE">
</uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE">
</uses-permission>
<uses-permission android:name="android.permission.INTERNET"> </uses-permission>
<uses-permission
android:name="com.android.launcher.permission.INSTALL_SHORTCUT"></uses-permissio
n>
```

**Step 3.**在 **AndroidManifest.xml** 中添加 **AdActivity**

AdActivity 是广告详情展示的载体，请在 AndroidManifest.xml 中添加 AdActivity:

```
<activity android:name="net.youmi.android.AdActivity"
android:configChanges="keyboard|keyboardHidden|orientation"/>
```

**Step 4.**初始化账号信息

在主 Activity 的 onCreate 中调用 AdManager.init() 初始化 App ID 、App Secret、请求广告间隔和测试模式等参数(请务必在任意 AdView 初始化前调用一次)。

```
//第一个参数为您的应用发布 Id
//第二个参数为您的应用密码
//第三个参数是请求广告的间隔，有效的设置值为30至200，单位为秒
//第四个参数是设置测试模式，设置为 true 时，可以获取测试广告，正式发布请设置此参数为 false
AdManager.init(Context context,String appid, String appsec, int intervalSecond, boolean isTestMode);
```

！ 注意:3.04版本开始 **AdManager.init** 方法的参数改为五个，加上了 **Context** 参数

！ 请将测试模式设置为 **false** 后上传至网站等待审核。

！ 未上传应用安装包、未通过审核的应用、模拟器运行，都只能获得测试广告，审核通过后，模拟器上依旧是测试广告，真机才会获取到正常的广告。

## Step 5.设置发布渠道号【可选】

有米广告条 3.02版以后的 sdk 加入了分发渠道号设置,以后可以统计来自各个分发渠道的数据。该设置为可选,当没有设置发布渠道号时，将采用默认渠道号0。

设置方法如下：把以下代码复制到 **AndroidManifest.xml** 文件中：

```
<meta-data android:name="YOUMI_CHANNEL" android:value="渠道编号"></meta-data>
```

注意:请在 **AndroidManifest.xml** 配置中加入 **meta-data** 参数，其中 **key** 为 **"YOUMI\_CHANNEL"** ,**value** 为 **int** 型的渠道编号，请为每个不同渠道号打一个独立的包。详细的渠道编号请参见: <http://wiki.youmi.net/wiki/PromotionChannelIDs>

## 1.2 常用布局方式

### 1.2.1 使用 xml 布局嵌入广告

#### Step 1.在布局 xml 中设置广告条容器

以下为一个实例：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <!--adViewLayout 为广告条容器-->
    <LinearLayout
        android:id="@+id/adViewLayout"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    >
    </LinearLayout>
</LinearLayout>
```

#### Step 2.XML 布局代码部分

```
public class XmlSample extends Activity {

    @Override
```

```

        protected void onCreate(Bundle savedInstanceState) {
            // TODO Auto-generated method stub
            super.onCreate(savedInstanceState);
            //          应用 Id          应用密码          广告请求间隔(s)    设置
测试模式[false 为发布模式]
            AdManager.init(this,"XXXXXXXXXXXX", "XXXXXXXXXXXX",          30,
false);
            LinearLayout adViewLayout = (LinearLayout) findViewById(R.id.adViewLayout);
            adViewLayout.addView(new AdView(this),
                new LayoutParams(LinearLayout.LayoutParams.FILL_PARENT,
LinearLayout.LayoutParams.WRAP_CONTENT));
        }
    }
}

```

### 1.2.2 使用代码布局嵌入广告

以下是一个示例

```

public class CodeSample extends Activity{

    /** Called when the activity is first created. */
    @Override protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        //          应用 Id          应用密码          广告请求间隔(s)    设置
测试模式[false 为发布模式]
        AdManager.init(this,"XXXXXXXXXXXX", "XXXXXXXXXXXX",          30,
false);
        LinearLayout layout=new LinearLayout(this);
        layout.setOrientation(LinearLayout.VERTICAL);
        layout.setBackgroundResource(R.drawable.bg);
        //初始化广告视图，可以使用其他的构造函数设置广告视图的背景色、透明度及字体颜色
        AdView adView = new AdView(this);
        LayoutParams params = new LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT);
        layout.addView(adView, params);
        setContentView(layout);
    }
}

```

### 1.2.3 使用悬浮布局嵌入广告

以下是一个示例

```

public class SuspensoidSample extends Activity{
    @Override protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub super.onCreate(savedInstanceState);
        //          应用 Id          应用密码          广告请求间隔(s)    设置
置测试模式[false 为发布模式]
        AdManager.init(this,"XXXXXXXXXXXX", "XXXXXXXXXXXX",          30,
false);
        setContentView(R.layout.main);
    }
}

```

```
        //初始化广告视图 AdView adView = new AdView(this);
        FrameLayout.LayoutParams params = new
FrameLayout.LayoutParams(FrameLayout.LayoutParams.FILL_PARENT,
FrameLayout.LayoutParams.WRAP_CONTENT);
        //设置广告出现的位置(悬浮于屏幕右下角)
        params.gravity=Gravity.BOTTOM|Gravity.RIGHT;
        //将广告视图加入 Activity 中 addContentView(adView, params);
    }
}
```

## 2.常见问题

### 2.1 版本更新内容

#### 3.0版

- 12 全新的广告条界面。
- 13 重新定义广告条尺寸，详情请查看文档。
- 14 支持应用程序版本更新功能。
- 15 重新定义 `AdListener` 接口为 `AdViewListener`，同时重命名其方法 `onReceiveAd` 为 `onAdViewSwitchedAd`。并且修改了 `AdView` 的 `setAdListener(AdListener adListener)` 方法为 `setAdViewListener(AdViewListener listener)`。详情请查看 API 文档。
- 16 为 `AdView` 类新增了一个方法：`refreshAd()`。调用该方法尝试刷新 `AdView` 展示的广告。在某些情况下，如想在一个停留 时间极短的界面上快速地显示广告条，可以调用该方法尝试刷新广告，这样可以最大可能地避免出现这样的情况：一个只停留十几秒的界面由于广告没有及时出来而 错过了展示广告的机会。详情请查看 API 文档
- 17 修改了 `AdManager` 类的 `init` 方法，取消了旧版本的 `appVersion` 参数，现在 `init` 方法只有4个参数。详情请查看 API 文档。
- 18 为 `AdManager` 新增了一个方法：`disableUpdateApp`。调用该方法可以取消使用应用程序版本更新功能。除非有特别需要，否则请不要调用此接口。详情请查看 API 文档
- 19 为 `AdManager` 新增了一个方法：`setDonotClearWebViewCache`。调用该方法设置 SDK 不删除所使用的 `WebView` 的缓存。除非有特别需要，否则请不要调用此接口。详情请查看 API 文档
- 20 修复了 SDK 内置网页浏览页在 `cmwap` 环境下出现乱码的一个 bug。
- 21 该版本 SDK 兼容系统版本1.6及以上，对于1.5版本的系统可能有一定的兼容性问题。

#### 3.01版

- 22 修复了3.0版导致混淆出错的 bug。
- 23 修复了未成功请求广告时广告条占据广告位的 bug。
- 24 轻量化了 `AdView` 的初始化流程，加快运行速度。

#### 3.02版

- 25 加入了分发渠道号设置,以后可以通过后台统计来自各个分发渠道的数据。
- 26 修复了对 `android 1.5`系统的兼容性问题。

#### 3.03版

- 27 修复了文字广告有时候出现显示样式混乱的一个 bug。
- 28 优化了请求广告线程的操作，减少资源消耗。

#### 3.04版

- 在广告条的宽度设置为 `FILL_PARENT` 的情况下可自适应容器控件的宽度。

#### 3.05版

- 优化了广告展示模式

### 3.07版

- 修复了初始化广告条设置 `setVisibility` 函数引起的bug

## 2.2 广告条的尺寸定义

新版本的 SDK 对广告条尺寸进行了调整:

**使用 `LayoutParams` 进行广告条尺寸的设置**

广告条的宽度必须设置为 `LayoutParams.FILL_PARENT` 或 `LayoutParams.WRAP_CONTENT`,规则如下:

当宽度设置为 `LayoutParams.FILL_PARENT` 时,广告条的宽度为屏幕当前状态的宽度。如 320\*480 的屏幕,当竖屏时,广告条宽度为 320px;当横屏时,广告条宽度为 480px。在 v3.04 以后,设置为该值时,广告条宽度与容器控件宽度一致

当宽度设置为 `LayoutParams.WRAP_CONTENT` 时,广告条的宽度为屏幕宽高中较小一边的值。如 320\*480 的屏幕,无论屏幕的方向如何,广告条的宽度都为 320px。

广告条的高度必须设置为 `LayoutParams.WRAP_CONTENT`,并且广告条的高度会自动调整为 50dp。

换算为 px,则:

对于 Low density (120), ldpi:

广告条的高度为 38px;

对于 Medium density (160), mdpi:

广告条的高度为 50px;

对于 High density (240), hdpi:

广告条的高度为 75px;

对于 Extra high density (320), xhdpi:

广告条的高度为 100px;

## 2.3 有米广告 Android SDK 使用哪种字符编码

有米广告 Android SDK 使用 UTF-8 字符编码,在嵌入广告的时候请使用 UTF-8 编程环境,否则会出现乱码情况。

## 2.4 有米广告 Android SDK 兼容 Android 系统 SDK 的哪些版本

有米广告 Android SDK 兼容 Android 系统 1.6 及以上版本 SDK,对于 1.5 及以下版本可能会有兼容性问题。

## 2.4 关于 `AdViewListener` 接口

当指定的 `AdView` 切换展示广告后或者请求广告失败后,该接口会通知监听者。

该接口的设计是为了方便那些有需要在广告条展示广告后进行一些界面处理的开发者,但它不适用于统计展示数,

因为在一个应用程序的生命周期中可能存在多个 **AdView** 实例，因此在运行的过程中，使用该接口统计到的切换展示广告的次数将会比广告的真正展示数多，如果依据该接口来统计展示数，结果将会与真实展示数相差很大。查询详细的展示数据请登录有米平台: <http://www.youmi.net>

ps:

旧版本该接口的原型是 **AdListener**，该接口的方法 **onReceiveAd** 实际上就是针对每个 **AdView** 实例切换广告的时候通知监听者，作用与新接口一样，但其命名不准确，导致了一些不必要的误会，现更正其命名为 **void onAdViewSwitchedAd(AdView adView)**。

## 2.5 关于广告没有成功展示的问题排查

如果出现广告没有展示的情况，请按下面的步骤检查:

29 检查广告配置是否正确，请务必仔细阅读并且按照"SDK 嵌入步骤"中的"准备工作"一节中的步骤来完成广告的参数配置。如果该参数配置正确仍然无法显示广告，请按下一步检查。

30 使用 **Eclipse** 开发的朋友可以通过查看 **DDMS** 的 **logcat** 输出，有米广告的 **Tag** 是 **YoumiSdk**。如果 **log** 信息中有输出 **Request Error Code**，请将错误码发给有米客服。

## 2.6 关于混淆 APK 出错的问题

如果使用 **xml** 布局并且使用 **proguard** 混淆 **apk**，请在 **proguard.cfg** 中加入以下代码:

```
-keep public class net.youmi.android.AdView
```

## 2.7 更多常见问题

请查看 <http://wiki.youmi.net>



## 3. 附录:API 文档

### 3.1 net.youmi.android.AdManager

#### Class Overview

有米广告参数配置管理类,用于初始化请求广告的相关设置

#### Summary

##### Public Methods

static	void	<a href="#">init(Context context,String appid, String appsec, int intervalSecond, boolean isTestMode)</a>	初始化广告配置
static	void	<a href="#">setDonotClearWebViewCache()</a>	SDK 在展示广告的过程中使用了 WebView,如果您希望保留 WebView 的缓存,可以调用此接口设置 SDK 不删除 WebView 缓存。如无特别需要,请不要调用此接口。 新版 SDK 加入了应用版本检查更新功能,如果您不希望使用应用程序检查更新功能时,可以调用此接口关闭自动检查更新功能。如无特别需要,请不要调用此接口。
static	void	<a href="#">disableUpdateApp()</a>	

#### Public Methods

**public static void init(Context context,String appid, String appsec, int intervalSecond, boolean isTestMode)**

初始化广告参数,必须在任意 AdView 实例化前调用。建议仅且在应用启动的时候在主 Activity 的 onCreate 中调用该方法初始化一次。

参数:

context

appid - 应用 Id

appsec - 应用密码

intervalSecond - 设置广告最小切换时间,单位为秒

isTestMode - 设置是否测试模式,如果设置为 true,可以在真机上展示测试广告。在正式发布的版本中应该置为 false

**public static void setDonotClearWebViewCache()**

SDK 中有使用到 WebView,为了避免应用程序中的缓存数据的积累造成大量的存储空间消耗,SDK 会自动清除使用到的 WebViewCache 缓存。如果您的应用程序中有使用 WebView,并且不希望清除缓存,可以调用此接口设置不清除缓存。否则,请不要调用此接口。

**public static void disableUpdateApp()**

有米广告 Android 3.0版 SDK 加入了应用程序自动检查更新功能，这样可以很方便您的应用升级。以后当您升级应用程序的时候，只需要将应用程序的最新版本上传到我们的服务器，审核通过以后，您的旧版应用程序在启动的时候就会自动检查更新，在得到用户的授权后，应用程序会下载最新版本安装，这样老用户可以更快地使用最新版本的软件。**SDK 默认开启应用程序检查更新功能** 如果您不希望使用该功能，可以调用此接口停用该功能。

## 3.2 net.youmi.android.AdView

### Class Overview

AdView 是展示广告的载体

### Summary

#### Public Constructors

[AdView\(Activity activity\)](#)

AdView 构造函数，使用默认样式。

[AdView\(Context context, AttributeSet attrs\)](#)

AdView 构造函数，用于 xml 布局

[AdView\(Context context, AttributeSet attrs, int defStyle\)](#)

AdView 构造函数，用于 xml 布局

[AdView\(Activity activity, int backgroundColor, int textColor, int backgroundTransparent\)](#)

AdView 的构造函数

#### Public Methods

int [getAdWidth\(\)](#)

获取广告条宽度

int [getAdHeight\(\)](#)

获取广告条高度

void [refreshAd\(\)](#)

尝试刷新广告

void [setAdViewListener\(AdViewListener listener\)](#)

设置广告条监听接口

### Public Constructors

**public AdView(Activity activity)**

AdView 构造函数，使用默认样式

**public AdView(Context context, AttributeSet attrs)**

AdView 构造函数，用于 xml 布局

**public AdView(Context context, AttributeSet attrs, int defStyle)**

AdView 构造函数，用于 xml 布局

**public AdView(Activity activity, int backgroundColor, int textColor, int backgroundTransparent)**

AdView 的构造函数

参数

activity - Activity

backgroundColor - 文字广告的背景颜色

textColor - 文字广告的字体颜色

backgroundTransparent - 文字广告的背景透明度,范围是0到255

## Public Methods

**public int getAdWidth()**

获取广告条宽度

**public int getAdHeight()**

获取广告条的高度

**public void refreshAd()**

在某些情况下，如停留时间较短的页面上放置广告条，希望使广告条尽快地展示广告，可以调用此接口尝试刷新展示广告。

**public void setAdViewListener(AdViewListener listener)**

设置广告条监听接口

## 3.3 net.youmi.android.AdViewListener

### Class Overview

AdView 监听接口，监听一个指定的 AdView 的切换广告的状态。

### Summary

Public Methods

void [onAdViewSwitchedAd\(AdView adView\)](#)

当指定的 AdView 实例切换所展示的广告时，会调用此接口通知监听者

void [onConnectFailed\(AdView adView\)](#)

当指定的 AdView 实例请求广告失败时，如遇到手机网络断开等情况，会调用此接口通知监听者

## Public Methods

**public void onAdViewSwitchedAd(AdView adView)**

当指定的 **AdView** 实例切换所展示的广告时，会通知此接口监听者。

该接口的设计是为了方便那些有需要在广告条展示广告后进行一些界面处理的开发者，但它不适用于统计展示数，因为在一个应用程序的生命周期中可能存在多个 **AdView** 实例，因此在运行的过程中，使用该接口统计到的切换展示广告的次数将会比广告的真正展示数多，如果依据该接口来统计展示数，结果将会与真实展示数相差很大。详细的展示数据请登录有米平台：<http://www.youmi.net> 查询。

ps: 旧版本该接口的原型是 **AdListener**，该接口的方法 **onReceiveAd** 实际上就是针对每个 **AdView** 实例切换广告的时候通知监听者，作用与新接口一样，但其命名不准确，导致了一些不必要的误会，现更正其命名为 **void onAdViewSwitchedAd(AdView adView)**。

**public void onConnectFailed(AdView adView)**

当指定的 **AdView** 实例请求广告失败时，如遇到手机网络断开等情况，会调用此接口通知监听者

## 3.4 net.youmi.android.AdActivity

### Class Overview

**AdActivity** 是展示广告详情的载体，请在 **AndroidManifest.xml** 中添加 **AdActivity**:

```
<activity android:name="net.youmi.android.AdActivity"
android:configChanges="keyboard|keyboardHidden|orientation"/>
```