

# 帷千动媒广告 SDK 安装手册

Android SDK v.3.1.3

# 目 录

1. 简介.....	2
2. 帷千广告 SDK 嵌入 .....	2
2.1 SDK 嵌入步骤 .....	2
2.2 Banner 广告使用 .....	5
2.3 推荐墙广告的使用 .....	6
2.4 插播广告的使用 .....	7
2.5 获取正式广告 .....	9
3. 远程管理.....	9
3.1 广告栏位申请 .....	9
3.2 管理广告栏位 .....	10
3.3 应用程序报表 .....	10
4. 常见问题.....	10
4.1 WQMobile Android SDK 兼容 Android 系统哪些版本.....	10
4.2 广告没有成功展示问题 .....	10
4.3 关于混淆 APK 出错的问题 .....	11



## 1. 简介

欢迎使用 WQMobile Android SDK。本着更方便开发者嵌入和使用帷千 SDK 的原则，WQMobile SDK 3.0 及以上版本在集成和使用方式上都有较大的变动，因此无论是否是第一次使用 WQMobile SDK，请仔细阅读以下内容。为了能够正常接收广告，请严格按照以下步骤嵌入 SDK。文档中的示例代码段都可以在 WQAdSample 工程中找到。

**注：**一个应用中只能使用唯一的 AccountKey，并且保证这个 Key 是您在官网上和应用包名绑定过的；不同类型的广告请在帷千官网申请不同的 AdSlot。

## 2. 帷千广告 SDK 嵌入

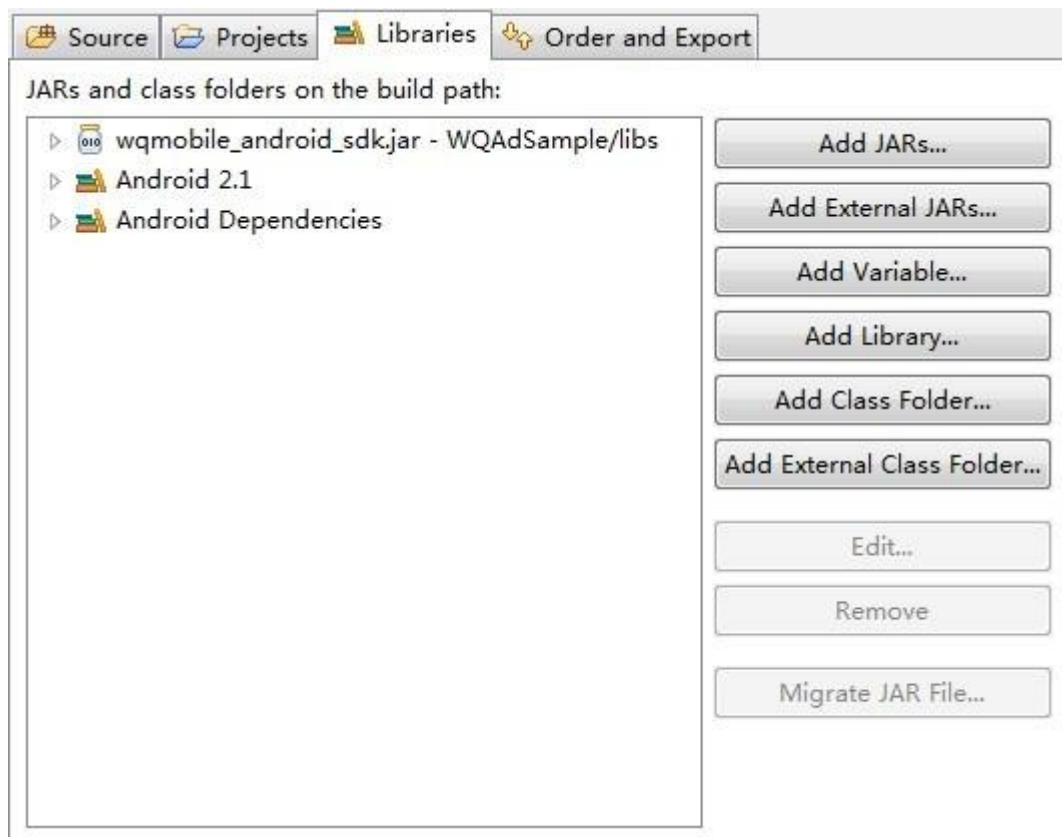
### 2.1 SDK 嵌入步骤

#### 2.1.1 添加 WQMobile Android SDK 到工程

首先在工程文件根目录下创建一个名为 libs 的子目录，并将 WQMobile SDK 的 JAR 包 (wqmobile\_android\_sdk.jar) 拷贝到 libs 目录下。

对于 Eclipse 工程，请参照下面的步骤添加 JAR 包：

- 1) 在“Package Explorer”页签中右击工程并选择“Properties”
- 2) 在左侧面板中选择“Java Build Path”
- 3) 在主窗口中选择“Libraries”页签
- 4) 点击“Add JARs...”按钮
- 5) 选择拷贝到 libs 目录下的 wqmobile\_android\_sdk.jar
- 6) 点击“OK”完成添加，如下图所示：



## 2.1.2修改 AndroidManifest.xml 文件

### 1) 添加 Activity 声明

在 AndroidManifest.xml 中的<application>标签中添加 Activity 声明(广告所需):

```
<activity
    android:name="com.wqmobile.sdk.WQActionHandler"
    android:configChanges="keyboardHidden|orientation|screenSize" />
<activity
    android:name="com.wqmobile.sdk.WQBrowser"
    android:configChanges="keyboardHidden|orientation|screenSize" />
```

### 2) 添加许可权限

在 AndroidManifest.xml 中的<manifest>标签中的<application>标签之前, 添加如下权限(五项必须添加, 否则将无法获取广告):



```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

另外，为了使富媒体广告更正常、更稳定地展示，建议同时添加以下权限：

```
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.VIBRATE" />
```

当完成以上的步骤设置后,请参考以下完整的 AndroidManifest.xml 范例

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.wqmobile.sdk.sample"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="7"
        android:targetSdkVersion="17" />

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

    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name" >
        <activity
            android:name="com.wqmobile.sdk.sample.WelcomeActivity"
            android:configChanges="keyboardHidden|orientation|screenSize" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
```



```
</activity>
<activity
    android:name=".CodeBannerAdActivity"
    android:configChanges="keyboardHidden/orientation/screenSize" />
<activity
    android:name=".XMLBannerAdActivity"
    android:configChanges="keyboardHidden/orientation/screenSize" />
<activity
    android:name=".InterstitialHalfAdActivity"
    android:configChanges="keyboardHidden/orientation/screenSize" />
<activity
    android:name="com.wqmobile.sdk.WQActionHandler"
    android:configChanges="keyboardHidden/orientation/screenSize" />
<activity
    android:name="com.wqmobile.sdk.WQBrowser"
    android:configChanges="keyboardHidden/orientation/screenSize" />

<meta-data
    android:name="WQ_CHANNEL"
    android:value="mm1" />
</application>

</manifest>
```

## 2.2 Banner 广告使用

出于安全和展示效果，建议通过代码的方式动态创建 WQMobileAdView。在代码中添加广告，设置 ADSLOT 和 ACCOUNT\_KEY：

```
public class CodeBannerAdActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.codebanner);
        setTitle(R.string.javacode_banner);

        RelativeLayout contentLayout = (RelativeLayout) findViewById(R.id.contentLayout);
```



```
WQAdView adView = new WQAdView(this);
LayoutParams params = new LayoutParams(LayoutParams.MATCH_PARENT,
LayoutParams.WRAP_CONTENT);
adView.setAdEventListener(new WQAdEventListener() {
    @Override
    public void onWQAdReceived(WQAdView adView) {
        Log.i(WelcomeActivity.TAG, "WQAdReceived");
    }

    @Override
    public void onWQAdFailed(WQAdView adView) {
        Log.i(WelcomeActivity.TAG, "WQAdFailed");
    }

    @Override
    public void onWQAdDismiss(WQAdView arg0) {
        Log.i(WelcomeActivity.TAG, "WQAdDismiss");
    }
});
adView.init(WelcomeActivity.ADSLOT_BANNER, WelcomeActivity.ACCOUNT_KEY);

contentLayout.addView(adView, params);
}
```

至此广告视图就成功的添加到应用程序中，当应用程序中放有广告视图的界面显示后，SDK 自动请求广告,成功获取广告后会自动出现在界面上。当该界面隐藏或销毁时，广告视图也随之停止请求和刷新，并随之销毁。

## 2.3 推荐墙广告的使用

推荐墙广告视图的添加有两种方式：

(1) 自定义推荐按钮， view 中添加点击事件的触发，设置 ADSLOT 和 ACCOUNT\_KEY：

```
WQAdView.openAdWall(this, ADSLOT, ACCOUNT_KEY);
```

当用户点击 view 触发事件时，就可以直接打开应用推荐墙界面。



(2) 帷千推荐按钮，添加方式和 Banner 类型广告视图一样，只需将 Banner 广告视图中 ADSLOT 和 ACCOUNT\_KEY 修改成推荐墙广告 ADSLOT 和 ACCOUNT\_KEY，就可以请求到帷千推荐按钮广告，点击推荐按钮，即可进入应用推荐墙界面。

## 2.4 插播广告的使用

**注意：**插播和 Banner 广告使用方式是不同的。插屏广告在使用的时候要先调用 loadInterstitialAd 方法请求到广告，在合适的时机调用 showInterstitialAd 方法来展示插屏广告；在成功展示的回调中调用 loadInterstitialAd 方法请求一次广告，便于在下次使用时直接通过 showInterstitialAd 方法来展示广告，减少一次广告请求时间。

插播广告分成两种类型：全屏插播和半屏插播。两种类型的广告区别：插播全屏是弹出式的全屏广告；插播半屏是嵌入式的广告，对应的广告尺寸有 300X250、600X500。您可以根据您的应用类型和使用场景选择对应的广告形式，具体使用方式参考以下示例

```
public class InterstitialHalfAdActivity extends Activity {

    private WQInterstitialAdView intersHalfAdView;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setTitle(R.string.inters_half);
        setContentView(R.layout.interstitial_half);
        intersHalfAdView = (WQInterstitialAdView) findViewById(R.id.intersHalfAdView);
        intersHalfAdView.init(WelcomeActivity.ADSLOT_INTERS_HALF,
WelcomeActivity.ACCOUNT_KEY);
        intersHalfAdView.setInterstitialAdListener(new WQInterstitialAdListener() {
            @Override
            //当插播广告被成功加载后回调，isAllFinished 为 true，表示所有的广告已经加载完成（一次
            广告加载可能会加载多条广告）
            public void onInterstitialAdRequestLoaded(boolean isAllFinished) {
                Log.i("InterstitialAd", "onInterstitialAdRequestLoaded(allFinished:"
                    + isAllFinished + ")");
                if (intersHalfAdView.isInterstitialAdReady() && isAllFinished)
                    intersHalfAdView.showInterstitialAd();
            }
        })

    @Override
```





```
//当插播广告要展示出来前, 回调该方法
public void onInterstitialAdPresent() {
    Log.i("InterstitialAd", "onInterstitialAdPresent");
}

@Override
//当插播广告加载失败后, 回调该方法
public void onInterstitialAdFailed() {
    Log.i("InterstitialAd", "onInterstitialAdFailed");
}

@Override
//当插播广告被关闭后, 回调该方法
public void onInterstitialAdDismiss() {
    Log.i("InterstitialAd", "onInterstitialAdDismiss");
}

@Override
//当插播广告展示成功后, 回调该方法
public void onWQAdView() {
    Log.i("InterstitialAd", "onWQAdView");
    //广告展示成功后, 重新请求广告, 减少一次请求广告时间
    intersHalfAdView.loadInterstitialAd();
}
});

//在需要展示插播广告之前, 先通过isInterstitialAdReady方法检查广告是否准备就绪
if (intersHalfAdView.isInterstitialAdReady()) {
    //展示广告
    intersHalfAdView.showInterstitialAd();
} else {
    //请求广告
    intersHalfAdView.loadInterstitialAd();
}

}

}
```

注:



1. 插播全屏广告只能通过代码初始化广告控件，而且广告初始化的时候，需要选择 WQAdMode 类型为 POPUP，详细可查看 WQAdSample 中例子。

```
intersFullAdView = new WQInterstitialAdView(this, WQAdMode.POPUP);  
intersFullAdView.init(ADSLT_INTERS_HALF, ACCOUNT_KEY);
```

2. 为更好兼容各种广告形式，需要在广告被隐藏或者遮挡时调用 pauseVideo 方法停止广告播放；在某些情况下需要暂停视频广告播放，当恢复到正常时，可以调用 continueVideo 方法让视频广告继续播放。

3. 插播类型部分广告资源可能较大，为广告顺利展示，需要尽早尽快地下载广告资源。SDK 包的 Sample 中的插播半屏广告提供了完整的插播广告使用流程，建议开发者查看并在自己的应用/游戏中实现该使用方法。

4. 插播视频广告在 Android 4.0 以下系统支持不是很完善，建议 4.0 及以上系统才获取该类型广告。

## 2.5 获取正式广告

在应用通过审核前，对应的正式 AccountKey 和 AdSlot 只能展示测试广告。请于当日 16 时之前提交应用程序，帷千工作人员会在 24 小时之内完成审核工作，审核通过之后，对应的正式 AccountKey 和 AdSlot 才够获取正常的付费广告。任何时候，都可以使用帷千公共测试 AccountKey 和 AdSlot 来测试帷千的广告效果及兼容性。使用公共测试 AccountKey 和 AdSlot 的应用在任何时候都只展示测试广告。

## 3. 远程管理

### 3.1 广告栏位申请

- 1) 登陆帷千官网 [www.wqmobile.com](http://www.wqmobile.com)，注册账户；
- 2) 登陆账户，进入“应用程序管理”页面，选择“新建应用程序”；
- 3) 填写应用程序信息（一个应用中只能使用唯一的 AccountKey，并且保证这个 Key 是在官网上和应用包名绑定过的）；
- 4) 设置广告位：点击“创建广告栏位”，填写广告栏位名称，选择广告类型，点击“保存并生成广告栏位 ID”，生成广告栏位对应 ID 等信息；将对应信息连同 SDK 植入至应用程序中并生成安装包；



- 5) 上传第（4）步中生成的安装包，等待审核；
- 6) 等待审核时可获取到测试广告，审核通过后方可获取付费广告。

注：除了横幅和推荐墙这两种广告类型之外，其他类型的广告嵌入请联系客服。

## 3.2 管理广告栏位

一个应用程序可以拥有一个或多个广告栏位，每个广告栏位都对应其唯一的 ID，每个账号对应一个发布 ID（即 KEY）。创建广告栏位并完成保存后将在每个广告栏位的信息中出现对应的 ID。

## 3.3 应用程序报表

可通过帷千平台上的应用程序报表查看应用程序广告播放的相关实时数据。具体操作流程：

- 1) 登陆帷千平台，选择“报表管理”中的“应用程序报表”；
- 2) 选择“创建新报表”，选填相关信息项保存后可查看所选的应用程序相关实时数据。

## 4. 常见问题

### 4.1 WQMobile Android SDK 兼容 Android 系统哪些版本

WQMobile Android SDK 兼容 Android 系统 2.1 及以上版本，对于 2.1 以下版本可能会有兼容性问题。

### 4.2 广告没有成功展示问题

在添加广告视图时，请避免以下情况，否则将影响广告的获取和展示：

- 1) 广告视图被其他任何视图遮挡；
- 2) 广告视图的范围在屏幕可视范围之外；
- 3) 广告视图为透明或隐藏；



- 4) 植入 v.3.0 及以上版本的 SDK，需重新获取相应的发布 ID，继续使用 v.3.0 以下版本 SDK 的发布 ID 将影响广告的正常获取。

### 4.3 关于混淆 APK 出错的问题

若使用 proguard 混淆 apk，请在 proguard.cfg 中加入以下代码，并保证项目 Project Build Target 的 API Level 为 17 以上：

---

```
-keepattributes *Annotation*, JavascriptInterface
```

```
-keep class com.wqmobile.** {  
    *;  
}
```

---

