

# 金山云移动广告SDK快速接入文档 V4.0.0

2018-1-8 V1.0.0 版本 - 汤洛

1.初版更新

## 目录

金山云移动广告SDK快速接入文档 V4.0.0

### 目录

- 1、文件清单
- 2、提供形式
- 3、快速集成
- 4、快速使用
  - 4.1、初始化及预加载
  - 4.2、展示广告
- 5、高级用法
  - 5.1、SDK配置项
  - 5.2、广告事件监听

## 1、文件清单

SDK包含的文件，如下所示：

- SDK demo工程
- jar包形式SDK，以及assets文件夹（包含SDK初始插件apk）
- aar形式SDK（包含SDK初始插件apk）
- SDK快速接入文档.pdf
- SDK接口说明文档.pdf

## 2、提供形式

SDK主要使用以下两种形式提供给客户：

- AAR文件形式（**推荐**）

- Jar包+Asset资源形式

客户可以根据自己工程的实际情况，任选其中一种方式进行集成即可

### 3、快速集成

#### SDK库文件导入

##### 方式一，基于AAR文件形式（推荐）

- 1、在客户的app module工程，根目录libs文件夹（如不存在，创建一个即可）下，放置对应的aar包。
- 2、在app module的build.gradle文件中，添加如下所示代码

```
1.  android {
2.      ...
3.      repositories {
4.          flatDir {
5.              dirs 'libs'
6.          }
7.      }
8.  }
9.
10. dependencies {
11.     //此处name字段中，应填写aar文件真实名称，这里仅以sdk-xxx.aar为例
12.     compile(name: 'sdk-xxx.aar', ext: 'aar')
13. }
```

### 3、添加Manifest权限

```
1.  <uses-permission
2.      android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
3.  <uses-permission
4.      android:name="android.permission.READ_EXTERNAL_STORAGE"/>
5.  <uses-permission
6.      android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
7.  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
8.  <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
9.  <uses-permission
10.     android:name="android.permission.ACCESS_NETWORK_STATE"/>
11.  <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
12.  <uses-permission android:name="android.permission.INTERNET"/>
```

```

9.     <uses-permission
10.         android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>
11.     <uses-permission
12.         android:name="android.permission.ACCESS_COARSE_LOCATION"/>
13.     <uses-permission
14.         android:name="android.permission.ACCESS_FINE_LOCATION"/>

```

#### 4、注册SDK内部组件

```

1.     <provider
2.         android:name="android.support.v4.content.FileProvider"
3.         android:authorities="${applicationId}.fileprovider"
4.         android:exported="false"
5.         android:grantUriPermissions="true">
6.         <meta-data
7.             android:name="android.support.FILE_PROVIDER_PATHS"
8.             android:resource="@xml/file_paths"/>
9.     </provider>

```

#### 5、将SDK文件目录下，res文件下xml目录，复制到工程的相应目录下,修改其中的包名，如下：

```

1.         //注意下方的path取值，需要填写用户自己的包名
2.         <external-path path="Android/data/com.xxx.xxx.xxx/"
3.             name="files_root" />
4.         <external-path path="cache/apk/." name="external_storage_root" />

```

#### 方式二，基于Jar包+Asset资源形式

1、在客户app module工程根目录libs文件夹（如不存在，创建一个即可）下，放置对应的jar包。

#### 2、添加Manifest权限

```

1.     <uses-permission
2.         android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
3.     <uses-permission
4.         android:name="android.permission.READ_EXTERNAL_STORAGE"/>
5.     <uses-permission
6.         android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
7.     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>

```

```

5.     <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
6.     <uses-permission
       android:name="android.permission.ACCESS_NETWORK_STATE"/>
7.     <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
8.     <uses-permission android:name="android.permission.INTERNET"/>
9.     <uses-permission
       android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>
10.    <uses-permission
       android:name="android.permission.ACCESS_COARSE_LOCATION"/>
11.    <uses-permission
       android:name="android.permission.ACCESS_FINE_LOCATION"/>

```

### 3、注册SDK内部组件

```

1.     <activity
2.         android:name="com.ksc.ad.sdk.ui.AdProxyActivity"
3.         android:theme="@android:style/Theme.Black.NoTitleBar.Fullscreen"
4.         android:configChanges="keyboardHidden|orientation|screenSize" />
5.     <activity
6.         android:name="com.ksc.ad.sdk.ui.AdPermissionProxyActivity"
7.         android:configChanges="keyboardHidden|orientation|screenSize"
8.
9.         android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
10.    />
11.    <service android:name="com.ksc.ad.sdk.service.AdProxyService" />
12.    <provider
13.        android:name="android.support.v4.content.FileProvider"
14.        android:authorities="${applicationId}.fileprovider"
15.        android:exported="false"
16.        android:grantUriPermissions="true">
17.        <meta-data
18.            android:name="android.support.FILE_PROVIDER_PATHS"
19.            android:resource="@xml/file_paths"/>
20.    </provider>

```

### 4、将SDK文件目录下，res文件下xml目录，复制到工程的相应目录下,修改其中的包名，如下：

```

1.         //注意下方的path取值，需要填写用户自己的包名
2.         <external-path path="Android/data/com.xxx.xxx.xxx/"
3.             name="files_root" />
4.         <external-path path="cache/apk/." name="external_storage_root" />

```

5、在project视图中选中app module < 右键 < open module setting, 在接下来的project structure视图中，左下角选中app module < dependency选项卡 < 点击左下角加号 < 选择jar dependency < 选择刚导入的jar包

6、将SDK文件目录下，assets文件夹中的内容，复制到客户app module中src/main/assets目录下

## 4、快速使用

### 4.1、初始化及预加载

建议在App启动后，第一个页面onCreate时进行SDK初始化及预加载操作

```
1.  class MainActivity extends Activity {
2.
3.      @Override
4.      protected void onCreate(Bundle savedInstanceState) {
5.          super.onCreate(savedInstanceState);
6.          KsyunAdSdk.getInstance().init(MainActivity.this, "your_app_id",
7. new IKsyunAdInitResultListener() {
8.              @Override
9.              public void onSuccess(Map<String, String> map) {
10.                  //SDK初始化成功,开始预加载
11.                  KsyunAdSdk.getInstance().preloadAd(new
12. IKsyunAdPreloadListener() {
13.                     @Override
14.                     public void onAdInfoSuccess() {
15.                         //加载广告配置信息成功
16.                     }
17.
18.                     @Override
19.                     public void onAdInfoFailed(final int errCode, final
20. String errMsg) {
21.                         //加载广告配置信息失败
22.                     }
23.
24.                     @Override
25.                     public void onAdLoaded(final String adSlotId) {
26.                         //该方法根据加载到的广告数量及广告位数量，可能会被多次调用
27.                     }
28.                 }
29.             }
30.         }
31.     }
32. }
```

```

26.         @Override
27.         public void onFailure(int errCode, String errMsg) {
28.             //SDK初始化失败处理
29.         }
30.     });
31. }
32.
33. }

```

## 4.2、展示广告

在广告位入口展示前，先调用hasAd方法判断当前广告位有无对应有效广告，根据结果决定是否展示入口。

广告展现后，待用户点击时，再调用showAd方法展示广告。

```

1.         KsyunAdSdk.getInstance().hasAd(adslot_id, new
IKsyunAdExistListener() {
2.             @Override
3.             public void onAdExist() {
4.                 //该广告位存在有效广告，则直接进入奖励视频广告。此处逻辑仅做示例
， 客户可根据实际情况自行修改
5.                 KsyunAdSdk.getInstance().showAd(this, adslot_id);
6.             }
7.
8.             @Override
9.             public void onNoAd(final int errCode, final String errMsg)
{
10.                 //该广告不存在有效广告，应隐藏该广告位奖励视频入口
11.             }
12.         });

```

## 5、高级用法

### 5.1、SDK配置项

在调用init初始化方法之前，可以通过设置SDK配置项，来进行环境及功能的可选配置。具体支持的配置项定义及说明详情，请参见SDK接口文档附录表。

```

1.         KsyunAdSdkConfig config = new KsyunAdSdkConfig();
2.         //设置SDK请求环境为测试环境
3.         config.setSdkEnvironment(KsyunAdSdkConfig.TEST_ENV);

```

```

4.         //设置奖励视频展示过程中，允许出现关闭按钮
5.         config.setShowCloseBtnOfRewardVideo(true);
6.         //设置奖励视频展示过程中，出现关闭按钮的时间点
7.         config.setCloseBtnComingTimeOfRewardVideo(5);
8.
9.         KsyunAdSdk.getInstance().init(MainActivity.this, appId, config,
new IKsyunAdInitResultListener() {
10.             @Override
11.             public void onSuccess(Map<String, String> map) {
12.
13.             }
14.
15.             @Override
16.             public void onFailure(int errCode, String errMsg) {
17.
18.             }
19.         });

```

## 5.2、广告事件监听

可以通过设置setAdListener接口，监听广告播放过程中用户对应的行为回调

```

1.     public interface IKsyunAdListener {
2.         //广告展示成功时回调
3.         void onShowSuccess(String adSlotId);
4.         //广告展示失败时回调
5.         void onShowFailed(String adSlotId, int errCode, String errMsg);
6.         //广告内容播放完成，一般用于视频类广告
7.         void onADComplete(String adSlotId);
8.         //广告被点击
9.         void onADClick(String adSlotId);
10.        //广告被关闭
11.        void onADClose(String adSlotId);
12.    }

```

对于奖励视频类型的广告，通过设置setRewardVideoAdListener接口，可以监听奖励条件是否达成回调。

```

1.     public interface IKsyunRewardVideoAdListener {
2.         //奖励条件达成
3.         void onAdAwardSuccess(String adSlotId);
4.
5.         //奖励条件未达成

```

```
6.     void onAdAwardFailed(String adSlotId, int errCode, String errMsg);  
7. }
```