



安智广告 SDK 接入文档

版本：2.2

目录

- 安智广告 SDK 接入文档..... 1
- 版本说明.....1
- 1 概述.....1
 - 1.1 主要内容.....1
- 2 接入说明.....1
 - 2.1 AndroidManifest 文件配置说明..... 1
 - 2.2 混淆配资.....2
 - 2.3 接入资源选择..... 4
 - 2.4 注意事项.....4
- 3 SDK 接口说明.....5
 - 3.1 开屏广告.....5
 - 3.2 banner 广告.....6
 - 3.3 插屏广告.....9
 - 3.4 原生广告.....11
 - 3.5 前贴片广告..... 13

● 版本说明

SDK 版本	内容	
2.2	安智广告 SDK 接入说明	

1 概述

1.1 主要内容

安智广告 SDK 集成多种广告形式给客户端集成，安智广告 SDK 集成形式有

- a、开屏广告
- b、插屏广告
- c、Banner 广告
- d、原生广告
- e、前贴片视频广告

2 接入说明

2.1 AndroidManifest 文件配置说明

Mainfest 文件配置说明参见 demo 或者 AndroidManifest.xml 文件主要包括权限、activity、service；

```
<meta-data android:name="ANZHI_AD_VERSIONS" android:value="2.2" ></meta-data>
```

如果 targetSdkVersion 设置值 ≥ 24 ，则需要在 application 标签中添加 provider 申明并在 res 中创建 xml 路径，并添加 demo 中 xml 下的所有文件

，否则不需

要添加。

```
<provider
```

```
    android:name="com.baidu.mobads.openad.FileProvider"
```

```
    android:authorities="${packageName}.bd.provider"
```

```
    android:exported="false"
```

```
    android:grantUriPermissions="true">
```

```
<meta-data
```

```
    android:name="android.support.FILE_PROVIDER_PATHS"
```

```
    android:resource="@xml/bd_file_paths" />
```

```
</provider>
```

```
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="${applicationId}.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">

<meta-data
    android:name="android.support.FILE_PROVIDER_PATHS"
    android:resource="@xml/gdt_file_path" />

</provider>
```

2.2 混淆配资

```
-keep public class com.leedavid.adslib.comm.**
-dontwarn com.leedavid.**
-keepclassmembers public class com.leedavid.adslib.comm.** { public *** ***(...);}
-keepclassmembers public class com.leedavid.adslib.comm.** { public <fields> ;}
-keep class com.qq.e.** { public protected *;}
-keep class android.support.v4.app.NotificationCompat** { public *;}
-keepclassmembers class * extends android.app.Activity { public void *(android.view.View);}
-keepclassmembers enum * { public static **[] values();
public static ** valueOf(java.lang.String);
}
-keep class com.baidu.mobads.*.** { *; }
-keep class com.afk.** { *;}
-keep class com.google.protobuf.** { *;} -keepattributes *Annotation* -keepattributes *JavascriptInterface*
-keep public class * implements com.afk.client.ads.inf.BaseListener
-keep public class com.afk.client.ads.inf.BaseListener
-keep public class * extends android.app.Activity
-keep public class * extends android.webkit.WebChromeClient
-keep public class com.anzhi.usercenter.sdk.AnzhiUserCenter
-keep class * implements com.anzhi.usercenter.sdk.BaseWebViewActivity$JsCallJavaInterface{*; }
-keep public class com.anzhi.sdk.ad.**{*;}
-keepclassmembers class com.afk.client.ads.inf.BaseListener {
    <fields>;
    <methods>;
}
```

```
}  
-keepclassmembers class com.anzhi.sdk.ad.control.MediaCallbackAz {  
    <fields>;  
    <methods>;  
}  
-keepclassmembers class com.anzhi.sdk.ad.control.GetPrerollInfoControl {  
    <fields>;  
    <methods>;  
}  
-keepclassmembers class com.anzhi.sdk.ad.manage.AnzhiAdPrerollAdCallBack {  
    public <fields>;  
    public <methods>;  
}  
-keepclassmembers class com.anzhi.sdk.ad.manage.AnzhiNativeAdCallBack {  
    public <fields>;  
    public <methods>;  
}  
-keepclassmembers class com.anzhi.sdk.ad.manage.AzMediaCallback {  
    public <fields>;  
    public <methods>;  
}  
-keepclassmembers class com.anzhi.sdk.ad.manage.AnzhiAdCallBack {  
    public <fields>;  
    public <methods>;  
}  
-keepclassmembers class * extends com.anzhi.sdk.ad.control.GetThrInfo{  
    public <fields>;  
    public <methods>;  
}  
-keepclassmembers class com.anzhi.sdk.ad.manage.AnzhiVideCallBack {  
    public <fields>;  
    public <methods>;  
}  
-keepclassmembers class * extends android.webkit.WebChromeClient{  
    public <fields>;
```

```

        public <methods>;
    }

-keepclassmembers class * extends com.anzhi.sdk.ad.main.AdBaseView{

    public <fields>;
    public <methods>;
}

-keepclassmembers class com.anzhi.sdk.ad.main.AdBaseView{

    public <fields>;
    public <methods>;
}

-keepclassmembers class * implements com.afk.client.ads.inf.BannerAdListener{

    public <fields>;
    public <methods>;
}

-keep class com.anzhi.usercenter.sdk.item.** {

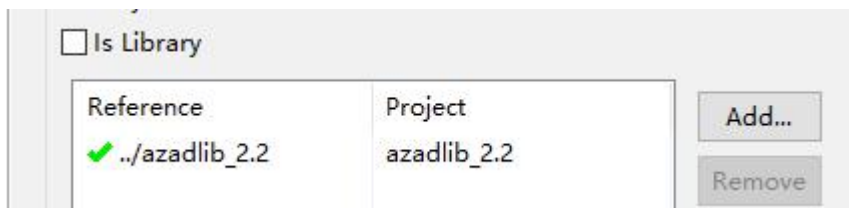
    <fields>;
    <methods>;
}

```

2.3 接入资源选择

资源包提供了 studio 和 eclipse 版本的 demo。

1. Eclipse 环境开发需要通过 IED 导入 DEMO 目录下 eclipse 目录的 lib 工程即可。如图：



2. studio 环境开发，需要在 build 文件增加配置

```

dependencies {
    compile fileTree(include: '*.jar', dir: 'libs')
    compile 'com.android.support:recyclerview-v7:23.4.0'
    compile project(':azadlib_2.2')
}

```

2.4 注意事项

- 1、包名和参数要对应，包名和参数要对应，包名和参数要对应，重要的事情说三遍！！
- 2、Studio 工程依赖 com.android.support:recyclerview-v7，eclipse 需要手动下载 com.android.support:recyclerview-v7 这个包放到 libs 目录下面，如果不需要使用 recyclerview 可以不加

3、目前支持的 SDK 版本，最低到 API 14（即 Android 4.0）：

如果您打包 App 时的 `targetSdkVersion >= 23`：请在先获取到 SDK 要求的所有权限，然后再调用 SDK 的广告接口。否则 SDK 将无法工作，我们建议您在 App 启动时就去获取。

如果您打包 App 时的 `targetSdkVersion >= 24`：除了需要处理好权限申请以外，还需要处理好文件访问的兼容性。

3 SDK 接口说明

SDK 接口说明包括开屏、banner、插屏、视频广告、本地原生广告、前贴片广告的 API 以及回调方法。

3.1 开屏广告

1. 开屏初始化方法

```
new AzSplashAd(Activity activity ,String appKey,String adid, AnzhiAdCallBack callBack, ,ViewGroup view,View skipView);
```

参数说明

参数	传值	说明
activity	Activity	开发者程序的开屏主 activity
appKey	String	开发者媒体的 ID 通安智广告后台申请获得
adid	String	开发者广告位 ID 通过安智广告后台申请获得，一个和 appKey 对应，一个 appKey 对应多个 adid;
callBack	AnzhiAdCallBack	广告通知回调接口
view	ViewGroup	加载开屏页面的父 view;
skipView	View	支持开发者自定义的跳过按钮（仅广点通有效）。SDK 要求 skipView 一定在传入后要处于 VISIBLE 状态，且其宽高都不得小于 3x3dp。

2. 加载开屏广告

```
public void loadAd();
```

3. 回调

```
new AnzhiAdCallBack() {  
    @Override  
    public void onShow() {  
        holder.setVisibility(View.INVISIBLE);  
        LogUtils.e("--闪屏广告展示--");  
    }  
    @Override
```

```
public void onReceiveAd() {}

@Override

public void onLoadFailed() {

    LogUtils.e("--闪屏广告加载失败---");

    finish();

    Intent intent = new Intent();

    intent.setClass(ASplashActivity.this, BannerActivity.class);

    ASplashActivity.this.startActivity(intent);

}

@Override

public void onCloseAd() {

    LogUtils.e("--闪屏广告关闭");

    finish();

    Intent intent = new Intent();

    intent.setClass(ASplashActivity.this, BannerActivity.class);

    ASplashActivity.this.startActivity(intent);

}

@Override

public void onAdExposure() {

}

@Override

public void onADTick(long millisUntilFinished) {

    tx.setText(String.format("点击跳过 %d", Math.round(millisUntilFinished / 1000f)));

}

@Override

public void onAdClick() {

    LogUtils.e("---闪屏广告被点击---");

}

}
```

3.2 banner 广告

1. Banner 初始化方法

```
new AzBannerAdView(Activity activity, String appKey, String adid, AnzhiAdCallBack mAdCallBack, ViewGroup
```

```
rootView);
```

参数说明

参数	传值	说明
activity	Activity	开发者程序的开屏主 activity
appKey	String	开发者媒体的 ID 通安智广告后台申请获得
adid	String	开发者广告位 ID 通过安智广告后台申请获得，一个和 appKey 对应，一个 appKey 对应多个 adid;
callBack	AnzhiAdCallBack	广告通知回调接口
rootView	ViewGroup	rootView 广告容器

2. 加载 banner 广告

```
public void loadAd();//
```

3. 设置仅对某些广告有效,设置 banner 宽度,不设置默认为屏幕宽度

```
public void setWidth(int w);//
```

4. 生命周期方法

生命周期方法需要在对应的 activity 生命周期方法中调用

```
void onDestroy();//
```

5. banner 回调

```
new AnzhiAdCallBack() {  
    @Override  
    public void onShow() {  
        LogUtils.e("---banner 广告展示---");  
    }  
  
    @Override  
    public void onReceiveAd() { // banner 广告忽略此回调  
    }  
  
    @Override  
    public void onLoadFailed() {  
        LogUtils.e("---banner 加载失败---");  
    }  
  
    @Override  
    public void onCloseAd() {  
        LogUtils.e("---banner 关闭---");  
    }  
  
    @Override
```



```
public void onAdExposure() {}

@Override
public void onADTick(long millisUntilFinished) { // banner 广告忽略此回调
}

@Override
public void onAdClick() {
    LogUtils.e("---banner 广告被点击---");
}
},rl);
```

3.3 插屏广告

1. 插屏广告初始化方法:

```
new InterstitialAdView(Activity activity,String appKey, String adid,AnzhiAdCallBack callBack);
```

参数	传值	说明
activity	Activity	开发者程序的主 activity
appKey	String	开发者媒体的 ID 通安智广告后台申请获得
adid	String	开发者广告位 ID 通过安智广告后台申请获得, 一个和 appKey 对应, 一个 appKey 对应多个 adid;
callBack	AnzhiAdCallBack	广告通知回调接口

2. 加载插屏广告

```
void loadAd();
```

3. 插屏广告是否加载完成

```
void isAdReady();
```

4. 展示插屏广告

```
void showAD();
```

5. 插屏广告的生命周期方法

```
void onDestroy();
```

6. 插屏回调

```
new AnzhiAdCallBack() {  
    /**  
     * 成功接受广告回调  
     */  
    @Override  
    public void onReceiveAd() {  
        Log.i("anzhi","成功接受广告回调");  
    }  
    /**  
     * 广告接受失败的回调,该值可能为空  
     */  
    @Override  
    public void onLoadFailed() {  
        Log.i("anzhi","广告接受失败的回调,该值可能为空");  
    }  
}
```

```
    }  
  
    /**  
     * 广告关闭通知  
     */  
  
    @Override  
    public void onCloseAd() {  
        Log.i("anzhi", " 广告关闭通知");  
    }  
  
    /**  
     * 广告点击通知  
     */  
  
    @Override  
    public void onAdClick() {  
        Log.i("anzhi", " 广告点击通知");  
    }  
  
    /**  
     * 广告展示通知  
     */  
  
    @Override  
    public void onShow() {  
        Log.i("anzhi", " 广告展示通知");  
    }  
  
    /**  
     * 广告出错回调  
     */  
  
    @Override  
    public void onAdExposure() {  
        Log.i("anzhi", " 广告出错回调");  
    }  
  
    @Override  
    public void onADTick(long millisUntilFinished) { //插屏广告忽略此回调  
    }  
  
});
```

3.4 原生广告

1. 原生广告初始化方法

```
new AzNativeExpressView(Activity activity, String appKey, String adId, AnzhiNativeAdCallBack mAdNativeCallBack, int azAdCount, int width, int height);
```

参数	传值	说明
activity	Activity	开发者程序的主 activity
appKey	String	开发者媒体的 ID 通安智广告后台申请获得
adid	String	开发者广告位 ID 通过安智广告后台申请获得, 一个和 appKey 对应, 一个 appKey 对应多个 adid;
mAdNativeCallBack	AnzhiNativeAdCallBack	广告通知回调接口
azAdCount	int	获取广告的条数 (取值范围 1-10)
width	int	广告区域的宽度, 单位 px, 默认值为-1
height	int	广告区域的高度, 单位 px, 默认值为-1

2. 原生广告加载方法

```
public void load();// 加载广告
```

```
bindView(ViewGroup container);//渲染曝光广告 nativeExpressADView.bindView(relativeLayout);
```

```
destroy();//释放占用的资源, 当不再展示广告时一定要调用这个方法去释放资源 nativeExpressADView.destroy()
```

```
new AnzhiNativeAdCallBack() {
    @Override
    public void onReceiveAd(NativeExpressViewData nativeExpressViewData) {
        LogUtils.e("---渲染广告成功---");
    }
    @Override
    public void onAdFail(String reason) {
        LogUtils.e("---加载广告失败---");
    }
    @Override
    public void onRenderFail(NativeExpressViewData nativeExpressViewData) {
        LogUtils.e("---渲染广告失败---");
    }
    @Override
    public void onCloseAd(NativeExpressViewData nativeExpressViewData) {
        // 当广告模板中的关闭按钮被点击时, 广告将不再展示。NativeExpressADView 也会被 Destroy, 不再可用。
    }
}
```

```
}  
  
@Override  
public void onAdClick(NativeExpressViewData nativeExpressViewData) {  
    LogUtils.e("---原生广告被点击---");  
}  
  
@Override  
public void onAdExposure(NativeExpressViewData nativeExpressViewData) {  
    LogUtils.e("--原生广告展示---");  
}  
  
@Override  
public void onADLoaded(List<NativeExpressViewData> nativeExpressViewData) {  
    // 释放前一个 NativeExpressADView 的资源  
    if (nativeExpressADView != null) {  
        nativeExpressADView.destroy();  
    }  
    if (relativeLayout.getVisibility() != View.VISIBLE) {  
        relativeLayout.setVisibility(View.VISIBLE);  
    }  
    if (relativeLayout.getChildCount() > 0) {  
        relativeLayout.removeAllViews();  
    }  
    nativeExpressADView = nativeExpressViewData.get(0);  
    // 保证 View 被绘制的时候是可见的，否则将无法产生曝光和收益。  
    nativeExpressADView.bindView(relativeLayout);  
}
```

3.5 前贴片广告

1. 前贴片广告初始化方法

```
new AzPrerollAd(Activity activity, String appKey, String adid, AnzhiAdPrerollAdCallBack mAdPrerollAdCallBack,
RelativeLayout rootView, int Adtype, AzMediaCallback azMediaCallback);
```

参数	传值	说明
activity	Activity	开发者程序的开屏主 activity
appKey	String	开发者媒体的 ID 通安智广告后台申请获得
adid	String	开发者广告位 ID 通过安智广告后台申请获得，一个和 appKey 对应，一个 appKey 对应多个 adid;
AnzhiAdPrerollAdCallBack	mAdPrerollAdCallBack	广告通知回调接口
rootView	RelativeLayout	广告展示
Adtype		指定百度广告类型， 可选项包括：PrerollAd.TYPE_NORMAL 普通前贴片广告）， PrerollAd.TYPE_NATIVE（原生前贴片广告）
azMediaCallback	AzMediaCallback	视频广告播放状态回调接口，专用于带有视频素材的广告对象

2. 前贴片广告加载方法

```
loadAd();
```

3. 设置广告的宽高， 只对原生前贴片广告有效（默认值为-1，单位 px）

```
setSize(int width, int height);
```

4. 前贴片广告回调方法

```
new AnzhiAdPrerollAdCallBack(){
    @Override
    public List<PrerollAdData> onAdLoaded(List<PrerollAdData> list) {
        Log.i(TAG, "onAdLoaded()");// 广告加载完成
        return null;
    };

    @Override
    public List<PrerollAdData> onADClicked(PrerollAdData prerollAdData) {
        Log.i(TAG, "前贴片点击");// 点击回调
        return null;
    }
}
```

```
@Override  
public List<PrerollAdData> onADVideoLoaded(PrerollAdData prerollAdData) {  
    return null;  
}  
  
@Override  
public void onAdFail(String sr) {  
    Log.i(TAG, "onAdFail()");  
}  
};
```

5. 视频广告回调

```
new AzMediaCallback() {  
    @Override  
    public void onVideoReady(long duration) { // 初始化完成  
        Log.i(TAG, "onVideoReady() + duration = " + duration);  
    }  
  
    @Override  
    public void onVideoStart() { // 开始播放前贴片视频广告  
        Log.i(TAG, "onVideoStart()");  
    }  
  
    @Override  
    public void onVideoPause() { // 暂停视频广告  
        Log.i(TAG, "onVideoPause()");  
    }  
  
    @Override  
    public void onVideoComplete() { // 视频播放结束或者手动到最后一帧  
        Log.i(TAG, "onVideoComplete()");  
    }  
  
    @Override  
    public void onReplayButtonClicked() { // SDK 中内置的重播按钮点击  
        Log.i(TAG, "onReplayButtonClicked()");  
    }  
  
    @Override  
    public void onADButtonClicked() { // SDK 内置的免费下载  
        Log.i(TAG, "onADButtonClicked()");  
    }  
}
```

```

@Override

public void onFullScreenChanged(boolean isFullScreen) { // 全屏回调 isFullScreen 为 true 表示进去
    全屏, false 表示退出全屏

    Log.i(TAG, "onFullScreenChanged() + isFullScreen = " + isFullScreen);

}

@Override

public void onAdFail(String reason) {

    Log.i(TAG, "onVideoStart() " + reason);

}

};

```

6. 前贴片视频方法

方法名	说明
String getAdLogoUrl()	获得 Logo 图 Url
String getImgUrl();	获得广告大图
int getAdType();	获得广告样式 (参见 demo)
boolean isAPP();	是否是 App 广告, 是则可下载
Void onExposured();	广告曝光时调用
void onClicked(View var1);	广告点击时调用, 必须在 onExposured();方法之后调用
void setMediaListener(MediaCallback var1);	设置视频广告监听器
void preLoadVideo();	预加载视频成功会调用 onADVideoLoaded
boolean isVideoLoaded();	判断视频素材是否完成
void play();	播放视频
void stop();	停止播放, 生命周期方法
boolean isPlaying();	是否正在播放视频
int getDuration();	判断视频是否静音
int getCurrentPosition();	获得当前的进度时间毫秒
void resume();	生命周期方法
void setVolumeOn(boolean var1);	Set 是否静音
AdContext getAdContext();	返回 AdContext 对象, 里面有和 activity生命周期对应的方法, 需和 activity一起调用

