

变现猫SDK集成指南-Android(v3.5.6)

1、接入前准备

[对接流程图](#)

[变现猫开发者后台注册流程](#)

[SDK下载地址：](#)

2、接入SDK

2.1 导入aar及SDK依赖包

[2.1.1 兼容性说明](#)

[2.1.2 oaid接入说明](#)

[2.1.2 崩溃日志收集说明](#)

2.2 添加权限

2.3 AndroidManifest.xml配置

2.4 代码混淆配置

3、SDK初始化

4、非标广告位

4.1 浮标广告

4.2 非标banner广告

4.3 自渲染（原生页面）

4.4 自渲染（H5页面）

[使用方法](#)

4.5 动画浮标广告

4.6 Tab广告

4.7 聚合图文广告

[自定义样式参数说明](#)

[MultiplePicTextStyle表](#)

[PicTextStyle表](#)

4.8 聚合非标Banner广告

[自定义样式参数说明](#)

[MultipleCustomBannerStyle表](#)

CustomBannerStyle表

*4.9 接入示例：[互动激励视频](#)

5、标准广告位

5.1 开屏广告

5.2 信息流

5.3 Banner广告

5.4 插屏广告

5.5 激励视频

5.6 沉浸式视频

5.7 全屏视频

*5.8 接入示例：[本地广告](#)

6. 绑定用户数据（可选）

7. 常见问题排查

7.1 日志过滤

7.2 Log提示：“广告位ID非法”

7.3 Log提示：“无广告”

7.4 Log提示：“please use GDTSDK in BXMSDK”

7.5 Log提示：“BxmSdk is not init, please check.”

7.7 Log提示：“本地播放视频回调为空”

7.8 活动页面，点击播放激励视频按钮，Toast提示：视频正在赶来，请重试

[修订历史](#)

1、接入前准备

接入广告SDK前，请您联系广告平台申请您的AppId，广告位id等。

对接流程图



变现猫开发者后台注册流程

https://fcdn.bianxianmao.com/download/sdk/docs/bxm_ssp_media_registration_process.pdf

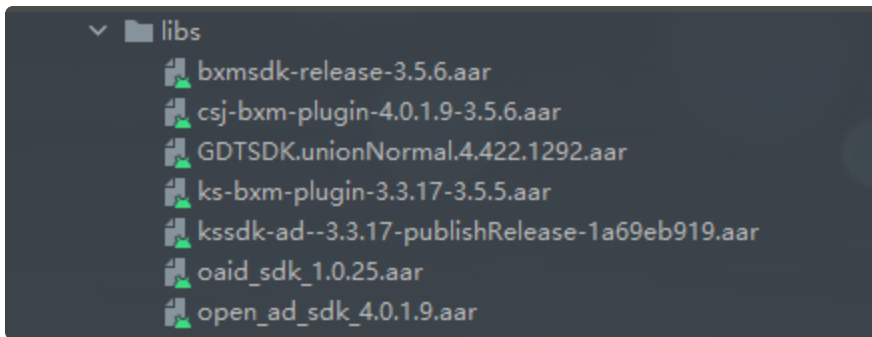
SDK下载地址：

https://fcdn.bianxianmao.com/download/sdk/bxm_sdk_android3.5.6_7068648194.zip

2、接入SDK

SDK接入推荐使用aar的方式进行接入，请解压提供的广告SDK，在压缩包中找到**bxmsdk-release-xxx.aar**。

接入过程中可参考压缩包中的SDKDemo进行使用，具体说明如下。



2.1 导入aar及SDK依赖包

将本 SDK 压缩包内的 **aar**文件 复制到 **App/libs** 文件夹(没有的话须手动创建), 并将以下代码添加到您 **App** 的**build.gradle**中

注：可选项，请开发同学与自己公司的商务、运营同学确认是否需要导入

```

1  repositories {
2      flatDir {
3          dirs 'libs'
4      }
5  }
6  dependencies {
7
8      //必须---请使用24.2.0及以上版本
9      implementation 'com.android.support:appcompat-v7:28.0.0'
10     //必须---变现猫SDK
11     implementation(name: 'bxmsdk-release-3.5.6', ext: 'aar')
12     //必须---广点通SDK, 最低支持版本为 4.370.1240
13     implementation(name: 'GDTSDK.unionNormal.4.422.1292', ext: 'aar')
14
15
16     //oaid移动安全联盟SDK---可选
17     implementation(name: 'oaid_sdk_1.0.25', ext: 'aar')
18
19     //穿山甲SDK及适配包---可选 (与自己公司的商务、运营同学确认是否需要导入)
20     implementation(name: 'open_ad_sdk_4.0.1.9', ext: 'aar')
21     implementation(name: 'csj-bxm-plugin-4.0.1.9-3.5.6', ext: 'aar')
22
23     //快手SDK及适配包---可选 (与自己公司的商务、运营同学确认是否需要导入)
24     implementation(name: 'kssdk-ad--3.3.17-publishRelease-1a69eb919', ext:
25     'aar')
26     implementation(name: 'ks-bxm-plugin-3.3.17-3.5.5', ext: 'aar')
27 }

```

2.1.1 兼容性说明

兼容性说明	
广点通	广点通当前版本为 4.422.1292，当前适配版本范围为 4.370.1240-4.422.1292；（如需其它版本请联系平台）
Android版本适配范围	minSdkVersion: 16, targetSdkVersion: 28,
android.support版本	为了适配下载和安装相关功能，在工程中引用的android.support版本，请使用24.2.0以及以上版本

2.1.2 oaid接入说明

接入原因：Android设备唯一标识包含IMEI，ANDROID_ID，Mac 地址等，随着国内外对隐私保护的越来越看重，这些唯一标识要么无法获取，要么就是无效值，在其他标识无法使用时，可以通过oaid做为设备唯一标识。

3.5.5版本起，SDK不在主动获取oaid，媒体可通过setOaid()方法传入。

获取oaid方法可参考如下代码：（示例使用版本为**1.0.25**，建议使用**>=1.0.25**的版本，其他版本存在兼容问题）

```
Java | 复制代码
1  public void getOaid() {
2      try {
3          MdidSdkHelper.InitSdk(this, true, new IIdentifierListener() {
4              @Override
5              public void OnSupport(boolean b, IdSupplier idSupplier) {
6                  if (idSupplier == null) {
7                      return;
8                  }
9
10                 BDAAdvanceConfig.getInstance().setOaid(idSupplier.getOaid());
11             });
12         } catch (Exception e) {
13             e.printStackTrace();
14         }
15     }
}
```

2.1.2 崩溃日志收集说明

SDK内部实现了Thread.UncaughtExceptionHandler接口，用于收集崩溃日志，为了获得更好的服务，建议开发者可采用如下步骤处理：

```
Java | 复制代码
1  //第一步： 获取系统默认的UncaughtExceptionHandler处理器
2  mDefaultHandler = Thread.getDefaultUncaughtExceptionHandler();
3  //第二步： 设置该CrashHandler为程序的默认处理器，并处理捕获异常的逻辑
4  Thread.setDefaultUncaughtExceptionHandler(this);
5  @Override
6  uncaughtException(Thread thread, Throwable ex){
7      APP处理逻辑
8  }
9  //第三步： 将捕获到的异常传递给系统默认异常处理器来处理
10 mDefaultHandler.uncaughtException(thread, ex);
}
```

2.2 添加权限

接入此SDK，需要相关权限，请在您的App的AndroidManifest.xml文件中，添加如下权限：

XML | 复制代码

```
1      <!--必要权限-->
2      <uses-permission android:name="android.permission.INTERNET" />
3      <uses-permission
4      android:name="android.permission.ACCESS_NETWORK_STATE" />
5      <uses-permission android:name="android.permission.READ_PHONE_STATE" />
6      <uses-permission
7      android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
8
9      <!--可选权限-->
10     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"
11     />
12     <uses-permission
13     android:name="android.permission.ACCESS_COARSE_LOCATION" />
14     <uses-permission
15     android:name="android.permission.ACCESS_FINE_LOCATION" />
```

注意：请求广告数据前，请务必申请IMEI权限（READ_PHONE_STATE），否则会造成部分广告无法正常下发，影响广告收入。所以强烈建议媒体在通过SDK请求广告前，先申请获取IMEI权限。

相关权限说明，见下表：

权限名称	用途	缺失导致问题	是否必须
READ_PHONE_STATE	设备信息：IMEI获取权限	无法获取设备标识IMEI。 严重影响广告填充率和广告效果	是
INTERNET	网络访问权限	无法访问网络。拉取不到广告	是
ACCESS_NETWORK_STATE	访问网络状态：检测当前网络状态是2G、3G、4G还是WiFi	无法获取网络状况。严重影响广告填充率	是
REQUEST_INSTALL_PACKAGES	应用安装权限	下载类广告无法安装。 严重影响广告价格 广告主根据安装情况进行效果评估	是
ACCESS_WIFI_STATE	设备信息：MAC获取权限	无法获取MAC地址。 严重影响广告填充率和广告效果	否
ACCESS_COARSE_LOCATION ACCESS_FINE_LOCATION	定位权限	无定位信息，影响广告填充率和定向推荐效果	否

2.3 AndroidManifest.xml配置

注意：在 AndroidManifest.xml 中的 Application 标签中添加 provider 标签

```
1 <application android:usesCleartextTraffic="true" >
2     //变现猫 v3.4.0新增
3     <provider
4         android:name="com.wgs.sdk.WgsFileProvider"
5         android:authorities="${applicationId}.wgs.fileprovider"
6         android:exported="false"
7         android:grantUriPermissions="true">
8         <meta-data
9             android:name="android.support.FILE_PROVIDER_PATHS"
10            android:resource="@xml/new_app_file_paths" />
11     </provider>
12
13     //广点通
14     <uses-library
15         android:name="org.apache.http.legacy"
16         android:required="false" />
17     <service
18         android:name="com.qq.e.comm.DownloadService"
19         android:exported="false" />
20     <activity
21         android:name="com.qq.e.ads.ADAActivity"
22         android:configChanges="keyboard|keyboardHidden|orientation|screenSize" />
23     <activity
24         android:name="com.qq.e.ads.PortraitADAActivity"
25         android:configChanges="keyboard|keyboardHidden|orientation|screenSize"
26         android:screenOrientation="portrait" />
27
28     // 穿山甲
29     <provider
30         android:name="com.bytedance.sdk.openadsdk.multipro.TTMultiProvider"
31         android:authorities="${applicationId}.TTMultiProvider"
32         android:exported="false" />
33
34     <provider
35         android:name="com.bytedance.sdk.openadsdk.TTFileProvider"
36         android:authorities="${applicationId}.TTFileProvider"
37         android:exported="false"
38         android:grantUriPermissions="true">
39         <meta-data
40             android:name="android.support.FILE_PROVIDER_PATHS"
41             android:resource="@xml/new_app_file_paths" />
42     </provider>
43 </application>
```


2.4 代码混淆配置

请确保您的应用打包混淆时，请在混淆配置文件添加如下配置：

Groovy | [复制代码](#)

```
1  #变现猫
2  -keep class com.wgs.sdk.**{ *;}
3  -keep class com.dhcw.sdk.**{*;}
4
5  -keep class com.qq.e.** {
6      public protected *;
7  }
8
9  -keep class android.support.v4.**{
10     public *;
11 }
12 #变现猫
13
14 #穿山甲 3900版本及以上版本混淆规则
15 -keep class com.bytedance.sdk.openadsdk.** { *; }
16 -keep class com.bytedance.frameworks.** { *; }
17
18 -keep class ms.bd.c.Pgl.**{*;}
19 -keep class com.bytedance.mobsec.metasec.ml.**{*;}
20
21 -keep class com.ss.android.**{*;}
22
23 -keep class com.bytedance.embedapplog.** {*;}
24 -keep class com.bytedance.embed_dr.** {*;}
25
26 -keep class com.bykv.vk.** {*;}
27 #穿山甲
28
29 #快手
30 -keep class org.chromium.** {*;}
31 -keep class org.chromium.** { *; }
32 -keep class aegon.chrome.** { *; }
33 -keep class com.kwai.**{ *; }
34 -dontwarn com.kwai.**
35 -dontwarn com.kwad.**
36 -dontwarn com.ksad.**
37 -dontwarn aegon.chrome.**
38 #快手
```

3、SDK初始化

请在您应用的 Application 的 onCreate() 方法中调用以下代码来初始化广告sdk。
具体可运行的代码参照：示例工程SDKDemo

```
Java | 复制代码
1 public class DemoApp extends Application {
2
3     @Override
4     public void onCreate() {
5         super.onCreate();
6         //必要：应用设置
7         BDAdvanceConfig.getInstance()
8             .setAppName("测试") // 1.设置应用名
9             .setCanUsePhoneState(true) // 2.是否允许获取设备信息，默认为true
10            .setCanUseLocation(true) // 3.是否允许使用位置信息，默认为true
11            .setCanUseAList(true) // 4.是否允许使用APPList，默认为true
12            .setOaid(getOaid()) // 5.传入oaid
13            .setDebug(true); // 6.如果需要，打开日志调试
14
15            // 必要：SDK初始化，appid 后台获取
16            BDManager.getStance().init(this,
17                "e4524c9e54e943d1adcda6c0bd711e8a");
18    }
19 }
```

4、非标广告位

活动入口根据SSP后台注册的广告位场景类型调用对应的SDK接口。

场景类型	SDK接口	重要方法说明
浮标	BDAdvanceFloatIconAd 参数说明： activity 当前页面 adContainer 展示广告容器，请尽量使用 FrameLayout adspotId 广告位ID SSP后台申请	setBdAdvanceFloatIconListener 广告回调 setBdAppNativeOnClickListener 活动内点击事件 setBdAdvanceCloseViewListener 点击浮标关闭按钮 loadAd 加载广告
按钮	BDAdvanceButtonAd	setBdAdvanceButtonListener 广告回调

非原生 banner	<p>BDAdvanceButtonAd</p> <p>参数说明:</p> <p>activity 当前页面</p> <p>adContainer 展示广告容器, 请尽量使用 FrameLayout</p> <p>adspotId 广告位ID SSP后台申请</p>	<p>setBdAdvanceButtonListener / 回调</p> <p>setBdAppNativeOnClickListener 活动内点击事件</p> <p>setBdAdvanceCloseViewListener 点击浮标关闭按钮</p> <p>loadAd 加载广告</p>
自渲染 (原生页面)	<p>BDAdvanceNativeRenderAd</p> <p>参数说明:</p> <p>activity 当前页面</p> <p>adspotId 广告位ID SSP后台申请</p>	<p>setBdAdvanceNativeRenderListener 广告加载回调</p> <p>loadAd 加载广告</p> <p>广告加载成功后需绑定广告入口组件</p> <p>mAdvanceNativeRenderItem.registerViewForInteraction 自渲染入口组件与广告数据绑定</p>
自渲染 (H5页面)	<p>BDAdvanceH5RenderAd</p> <p>参数说明:</p> <p>activity 当前页面</p> <p>mWebView WebView组件</p>	<p>onUrlLoading 监听页面 url</p>
动画浮标	<p>BDAdvanceAnimationFloatIconAd</p> <p>参数说明:</p> <p>activity 当前页面</p> <p>adContainer 展示广告容器, 请尽量使用 RelativeLayout</p> <p>adspotId 广告位ID SSP后台申请</p>	<p>setBdAdvanceAnimationFloatIconListener 广告回调</p> <p>setBdAppNativeOnClickListener 活动内点击事件</p> <p>setBdAdvanceCloseViewListener 点击浮标关闭按钮</p> <p>loadAd 加载广告</p>
Tab	<p>BDAdvanceTabAd</p> <p>参数说明:</p> <p>activity 当前页面</p> <p>adspotId 广告位ID SSP后台申请</p>	<p>setBdAdvanceListener 广告加载回调</p> <p>tabAdItem.setAdListener 广告曝光、点击回调</p>
聚合图文	<p>BDAdvanceMultiplePicTextAd</p> <p>参数说明:</p> <p>context 上下文</p> <p>adContainer 展示广告容器 (容器宽高请按以下方式配置:</p>	<p>setBdAdvanceMultiplePicTextListener 广告加载回调</p> <p>setMultiplePicTextStyle 设置广告样式</p>

	<p><code>android:layout_width="match_parent"</code> <code>android:layout_height="wrap_content"</code> <code>)</code> adspotId 广告位ID SSP后台申请</p>	<p>loadAds 加载聚合广告</p> <p>destroyAd 销毁广告</p>
聚合非标 Banner	<p>BDAvanceMultipleCustomBannerAd</p> <p>参数说明:</p> <p>context 上下文</p> <p>adContainer 展示广告容器 (容器宽高请按以下方式配置: <code>android:layout_width="match_parent"</code> <code>android:layout_height="wrap_content"</code> <code>)</code> adspotId 广告位ID SSP后台申请</p>	<p>setBdAvanceMultipleCustomBannerListener 广告加载回调</p> <p>setMultipleCustomBannerStyle 设置广告样式</p> <p>loadAds 加载聚合广告</p> <p>destroyAd 销毁广告</p>

4.1 浮标广告

浮标广告是非标广告的一种，以图片方式展示广告入口，提供静图和动图样式，支持点击刷新、定时刷新模式。

```
1 //1创建浮标广告
2 //adContainer 展示广告容器, 请尽量使用FrameLayout
3 //adspotId 广告位ID 变现猫后台申请
4 BDAdvanceFloatIconAd bdAdvanceFloatIconAd
5 = new BDAdvanceFloatIconAd(Activity activity, ViewGroup adContainer,
6 String adspotId);
7 //2.设置回调
8 bdAdvanceFloatIconAd.setBdAdvanceFloatIconListener(new BDAdvanceFloatIconL
9 istener() {
10     @Override
11     public void onActivityClosed() {
12         //活动页关闭
13     }
14     @Override
15     public void onAdShow() {
16         //广告展示
17     }
18     @Override
19     public void onAdFailed() {
20         //广告加载失败
21     }
22     @Override
23     public void onAdClicked() {
24         //广告点击
25     }
26 });
27 //互动激励视频, 具体可参考示例代码 或 《互动激励视频》接入部分说明
28 bdAdvanceFloatIconAd.setBdAppNativeOnClickListener(new
29 BDAppNativeOnClickListener() {
30     @Override
31     public void onActivityClosed() {
32     }
33     @Override
34     public void onClick(int type, String adid) {
35         //type=1 加载视频 type=2 播放视频
36         if(type==1){
37             //加载视频 绑定激励视频回调
38             TestPlayVideo.getInstance()
39                 .load(FloatIconActivity.this, bdAdvanceFloatIconAd);
40         }else if(type==2){
41             //播放视频
42             TestPlayVideo.getInstance()
43                 .play(FloatIconActivity.this);
44         }
45     }
46 });
47 //后台可配置是否打开关闭功能, 打开时, 点击关闭按钮触发该回调
```

```
45     bdAdvanceFloatIconAd.setBdAdvanceCloseViewListener(new
BDAdvanceCloseViewListener() {
46         @Override
47         public void onClose() {
48
49         }
50     });
51     //3.加载广告
52     bdAdvanceFloatIconAd.loadAd();
53
54     @Override
55     protected void onDestroy() {
56         super.onDestroy();
57         if (bdAdvanceFloatIconAd != null) {
58             bdAdvanceFloatIconAd.destroyAd();
59             bdAdvanceFloatIconAd = null;
60         }
61     }
```

4.2 非标banner广告

非标banner广告是非标广告的一种，类似Button按钮，以图片方式展示广告入口，提供静图和动图样式，支持点击刷新、定时刷新模式。

```
1 //1.创建非标banner广告
2 //adContainer 展示广告容器
3 //adspotId 广告位ID 变现猫后台创建
4 bdAdvanceButtonAd = new BDAdvanceButtonAd(Activity activity, ViewGroup
adContainer, String adspotId);
5 //2.设置回调
6 bdAdvanceButtonAd.setBdAdvanceButtonListener(new BDAdvanceButtonListener()
{
7     @Override
8     public void onActivityClosed() {
9         //活动页关闭
10    }
11    @Override
12    public void onAdShow() {
13        //广告展示
14    }
15    @Override
16    public void onAdFailed() {
17        //广告加载失败
18    }
19    @Override
20    public void onAdClicked() {
21        //广告点击
22    }
23 });
24 //互动激励视频, 具体可参考示例代码 或 《互动激励视频》接入部分说明
25 bdAdvanceButtonAd.setBdAppNativeOnClickListener(new
BDAppNativeOnClickListener() {
26     @Override
27     public void onActivityClosed() {
28     }
29
30     @Override
31     public void onClick(int type, String adid) {
32         //type=1 加载视频 type=2 播放视频
33         if(type==1){
34             //加载视频 绑定激励视频回调
35             TestPlayVideo.getInstance()
36                 .load(ButtonAdActivity.this, bdAdvanceButtonAd);
37         }else if(type==2){
38             //播放视频
39             TestPlayVideo.getInstance()
40                 .play(ButtonAdActivity.this);
41         }
42     }
43 });
44 //后台可配置是否打开关闭功能, 打开时, 点击关闭按钮触发该回调
```

```
45  bdAdvanceButtonAd.setBdAdvanceCloseViewListener(new
    BdAdvanceCloseViewListener() {
46      @Override
47      public void onClose() {
48      }
49  });
50  //3.加载广告
51  bdAdvanceButtonAd.loadAd();
```

4.3 自渲染（原生页面）

自渲染（原生页面）是非标广告的一种，为Android原生页面提供灵活的入口渲染方式，开发者调用自渲染广告接口获取广告素材后，可直接填充，也可以使用自有素材展示广告入口，并绑定入口组件的点击事件。接入难度高于浮标和按钮广告。


```
1  /**
2   * 自渲染
3   */
4  public class DispatchNativeRenderAdActivity extends Activity implements
BDAdvanceNativeRenderListener, View.OnClickListener {
5      private BDAdvanceNativeRenderItem bdAdvanceNativeRenderItem;
6      private FrameLayout advanceNativeAdContainer;
7      private Button feedButton;
8
9      @Override
10     protected void onCreate(@Nullable Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_dispatch_mative_render_ad);
13         advanceNativeAdContainer =
findViewById(R.id.advance_native_ad_container);
14
15         findViewById(R.id.bt_load_express_ad).setOnClickListener(this);
16         feedButton = findViewById(R.id.bt_feed_ad);
17         BDManger.getStance().requestPermission(this);
18
19     }
20
21     @Override
22     public void onClick(View view) {
23         if (view.getId() == R.id.bt_load_express_ad) {
24             loadAd();
25         }
26     }
27
28     private void loadAd() {
29         BDAdvanceNativeRenderAd bdAdvanceFeedAd = new
BDAdvanceNativeRenderAd(this, advanceNativeAdContainer,
Constants.FEED_AD_SPOT_ID);
30         bdAdvanceFeedAd.setBdAdvanceNativeRenderListener(this);
31         bdAdvanceFeedAd.loadAD();
32     }
33
34     @Override
35     protected void onResume() {
36         super.onResume();
37         if (bdAdvanceNativeRenderItem != null) {
38             // 必须要在Actiivty.onResume()时通知到广告数据, 以便重置广告恢复状态
39             bdAdvanceNativeRenderItem.resume();
40         }
41     }
42
43     @Override
```

```

44     public void onLoadAd(List<BDAdvanceNativeRenderItem>
advanceFeedItemList) {
45         bdAdvanceNativeRenderItem = advanceFeedItemList.get(0);
46         //使用接口返回的素材，也可以使用自有素材
47         setFeedAd(bdAdvanceNativeRenderItem);
48         ToastUtils.showToast(getApplicationContext(), "广告数据加载成功");
49         bindListener(bdAdvanceNativeRenderItem);
50     }
51
52     private void bindListener(final BDAdvanceNativeRenderItem
bdAdvanceFeedItem) {
53         List<View> clickViewList = new ArrayList<>();
54         clickViewList.add(feedButton);
55         // advanceNativeAdContainer 显示view的父窗口，用于曝光统计
56         // clickViewList 入口组件，用于绑定点击事件
57         // AdInteractionListener 广告回调
58         bdAdvanceFeedItem.registerViewForInteraction(advanceNativeAdContainer,
clickViewList, new AdInteractionListener() {
59             @Override
60             public void onAdClicked(View view) {
61                 Toast.makeText(getApplicationContext(), "onAdClicked",
Toast.LENGTH_SHORT).show();
62             }
63
64             @Override
65             public void onAdShow() {
66                 Toast.makeText(getApplicationContext(), "onAdShow",
Toast.LENGTH_SHORT).show();
67             }
68         });
69         //如果不需要本地播放视频，请不要设置
70         bdAdvanceFeedItem.registerBdAppNativeOnClickListener(new
BDAppNativeOnClickListener() {
71
72             @Override
73             public void onActivityClosed() {
74
75             }
76
77             @Override
78             public void onClick(int type, String s) {
79                 //type=1 加载视频    type=2 播放视频
80                 if(type==1){
81                     //加载视频 绑定激励视频回调
82                     TestPlayVideo.getInstance().load(DispatchNativeRenderAdActivity.this,
(BDAdvanceBaseAppNative) bdAdvanceFeedItem);
83                     }else if(type==2){
84                         TestPlayVideo.getInstance().play(DispatchNativeRenderAdActivity.this);
85                     }
86             }
87         });
88     }

```

```

85
86     }
87 });
88 }
89
90 @Override
91 public void onAdFailed() {
92 }
93
94 private void setFeedAd(BDAdvanceNativeRenderItem bdAdvanceFeedItem) {
95     advanceNativeAdContainer.removeAllViews();
96
97     View inflateView =
LayoutInflater.from(this).inflate(R.layout.layout_native_express_view_one,
advanceNativeAdContainer);
98
99     ImageView ivExpressAd =
inflateView.findViewById(R.id.bxm_iv_express_ad);
100     TextView tvExpressTitle =
inflateView.findViewById(R.id.bxm_tv_express_title);
101     TextView tvExpressSubTitle =
inflateView.findViewById(R.id.bxm_tv_express_subtitle);
102     Glide.with(this)
103         .load(bdAdvanceFeedItem.getImageList().get(0))
104         .into(ivExpressAd);
105     tvExpressTitle.setText(bdAdvanceFeedItem.getDescription());
106     tvExpressSubTitle.setText(bdAdvanceFeedItem.getTitle());
107
108 }
109
110 @Override
111 protected void onDestroy() {
112     super.onDestroy();
113     if (bdAdvanceNativeRenderItem != null) {
114         // 必须要在Activity.destroy()时通知到广告数据，以便释放内存
115         bdAdvanceNativeRenderItem.destroy();
116     }
117 }
118 }

```

4.4 自渲染（H5页面）

自渲染（H5页面）是非标广告的一种，为Android内H5页面提供灵活的入口渲染方式，开发者调用自渲染广告JS接口获取广告素材后，可直接填充，也可以使用自有素材展示广告入口，并绑定入口组件的点击事件。接入难度高于自渲染（原生页面）广告。

H5页面接入分两部分：**Android原生WebView部分**、**JS部分**。

一、Android原生WebView部分

Java | [复制代码](#)

```
1  mWebView.setWebViewClient(new CustomWebViewClient());
2  BDAdvanceH5RenderAd bdAdvanceH5RenderAd = new
   BDAdvanceH5RenderAd(DispatchH5RenderAdActivity.this, mWebView);
3
4  private class CustomWebViewClient extends WebViewClient {
5      @Override
6      public boolean shouldOverrideUrlLoading(WebView view, String url) {
7          if(bdAdvanceH5RenderAd==null){
8              bdAdvanceH5RenderAd = new
   BDAdvanceH5RenderAd(DispatchH5RenderAdActivity.this, mWebView);
9          }
10         bdAdvanceH5RenderAd.onUrlLoading(url);
11         return true;
12     }
13     @Override
14     public void onPageFinished(WebView view, String url) {
15         super.onPageFinished(view, url);
16     }
17     @Override
18     public void onReceivedSslError(WebView view, SslErrorHandler handler,
   SslError error) {
19         handler.proceed();
20     }
21 }
```

二、JS部分

`window.bxmAdsJsSdk.initBxmSdk({ adspotId, el, sR, success, error})`

参数名	类型	是否必传	描述
adspotId	str	Y	广告位id
el	str	Y	绑定的元素节点，可以是class、id等；如.bxm、#bxm
sR	bool	N	是否采用自渲染（默认: false） true: 采用自己写的样式渲染 false: 采用sdk样式渲染
success(parameters)	function	N	初始化成功回调 返回参数: {adspotId: {img, title, content}} adspotId: 请求传入的广告位id img: 图片地址 title: 标题 content: 内容（对应副标题）
error	function	N	初始化失败回调
onClickBack	function	N	点击节点回调
type	str	N	渲染类型 2:上图下文, 3: 图

使用方法

注: js不能加载多次, 封装时使用单例模式, 建议使用方法一和方法二

方法一: 直接引入

JavaScript | [复制代码](#)

```

1  <script type="text/javascript"
   src="//cdn.ituibei.cn/JSSDK/bxmAdsJsSdk.js"></script>
2  <script>
3
4    //初始化
5    window.bxmAdsJsSdk.initBxmSdk({adspotId: '804827005014', el: '#bxm',
   sR: false, type:3, success: function() {}, error: function() {},
   onClickBack: function() {}});
6
7  </script>

```

方法二: promise封装

```
1 function bxmLoadJs(url) {
2     return new Promise(resolve => {
3         var head = document.getElementsByTagName('head')[0];
4         var script = document.createElement('script');
5         script.type = 'text/javascript';
6         script.src = url;
7         script.onload = script.onreadystatechange = function () {
8             if (!this.readyState || this.readyState === "loaded" ||
this.readyState === "complete"){
9                 resolve();
10                script.onload = script.onreadystatechange = null;
11            }
12        };
13        head.appendChild(script);
14    });
15 }
16
17 //调用一次保证js不多次加载
18 var bxmAdsSdk = bxmLoadJs('//cdn.ituibe.cn/JSSDK/bxmAdsJsSdk.js')
19
20 bxmAdsSdk.then(() => {
21     window.bxmAdsJsSdk.initBxmSdk({
22         adspotId: '804827005014',
23         el: '#bxm', sR: false, type:3,
24         success: function() {},
25         error: function() {},
26         onClickBack: function() {}
27     });
28 })
29 bxmAdsSdk.then(() => {
30     window.bxmAdsJsSdk.initBxmSdk({
31         adspotId: '*****',
32         el: '#bxm', sR: false, type:3,
33         success: function() {},
34         error: function() {},
35         onClickBack: function() {}
36     });
37 })
```

方法三：js引入

```
1 <script>
2   function bxmLoadJs(url, callback) {
3     var head = document.getElementsByTagName('head')[0];
4     var script = document.createElement('script');
5     script.type = 'text/javascript';
6     script.src = url;
7     if(typeof callback === 'function'){
8       script.onload = script.onreadystatechange = function () {
9         if (!this.readyState || this.readyState === "loaded" ||
10        this.readyState === "complete"){
11           callback();
12           script.onload = script.onreadystatechange = null;
13         }
14       };
15     head.appendChild(script);
16   }
17   bxmLoadJs('//cdn.ituibeicn.com/JSSDK/bxmAdsJsSdk.js', function() {
18     //初始化
19     window.bxmAdsJsSdk.initBxmSdk({
20       adspotId: '804827005014',
21       el: '#bxm', sR: false, type:3,
22       success: function() {},
23       error: function() {},
24       onClickBack: function() {}
25     });
26   })
27 </script>
```

方法四：js单例封装

```
1 function bxmLoadJs(url) {
2     var pending = 0;
3     var fulfilled = 0;
4     var callbacks = [];
5
6     var useCallbacks = function(callback) {
7         if(typeof callback === 'function') {
8             callbacks.push(callback);
9         }
10
11         if(pending === 1 && fulfilled === 1) {
12             var len = callbacks.length;
13             for(var i=0; i < len; i++) {
14                 if(typeof callbacks[i] === 'function') {
15                     callbacks[i]();
16                 }
17             }
18             callbacks = [];
19         }
20
21         if(pending === 1) return true;
22     };
23
24     return function(callback) {
25         if(useCallbacks(callback)) return;
26         var head = document.getElementsByTagName('head')[0];
27         var script = document.createElement('script');
28         script.type = 'text/javascript';
29         script.src = url;
30         pending = 1;
31         script.onload = script.onreadystatechange = function () {
32             if (!this.readyState || this.readyState === "loaded" ||
33 this.readyState === "complete"){
34                 fulfilled = 1;
35                 useCallbacks();
36                 script.onload = script.onreadystatechange = null;
37             }
38         };
39         head.appendChild(script);
40     };
41
42     //调用一次保证js不多次加载
43     var bxmAdsSdk = bxmLoadJs('//cdn.ituibe.cn/JSSDK/bxmAdsJsSdk.js');
44
45     bxmAdsSdk(() => {
46         window.bxmAdsJsSdk.initBxmSdk({
47             adspotId: '804827005014',
```



```
48     el: '#bxm', sR: false, type:3,  
49     success: function() {},  
50     error: function() {},  
51     onClickBack: function() {}  
52   });  
53 });  
54 bxmAdsSdk(() => {  
55   window.bxmAdsJsSdk.initBxmSdk({  
56     adspotId: '*****',  
57     el: '#bxm', sR: false, type:3,  
58     success: function() {},  
59     error: function() {},  
60     onClickBack: function() {}  
61   });  
62 });
```

4.5 动画浮标广告

动画浮标广告在浮标广告的基础上增加了动画播放，加载广告后会先展示一段动画，然后停止并展示浮标广告。

```
1 //adContainer 展示广告容器, 请尽量使用RelativeLayout,
2 //请按照标准尺寸 (650*650, 120*650, 450*120 (单位: PX) ) 设置大小
3 //adspotId 广告位ID 变现猫后台申请
4 BDAdvanceAnimationFloatIconAd bdAdvanceAnimationFloatIconAd
5 = new BDAdvanceAnimationFloatIconAd(Activity activity, ViewGroup
6 adContainer, String adspotId);
7 //设置回调
8 bdAdvanceAnimationFloatIconAd.setBdAdvanceAnimationFloatIconListener(new
9 BDAdvanceAnimationFloatIconListener() {
10     @Override
11     public void onActivityClosed() {
12         //活动页关闭
13     }
14     @Override
15     public void onAdShow() {
16         //广告展示
17     }
18     @Override
19     public void onAdFailed() {
20         //广告加载失败
21     }
22     @Override
23     public void onAdClicked() {
24         //广告点击
25     }
26 });
27 //互动激励视频, 具体可参考示例代码 或 《互动激励视频》接入部分说明
28 bdAdvanceAnimationFloatIconAd.setBdAppNativeOnClickListener(new
29 BDAppNativeOnClickListener() {
30     @Override
31     public void onActivityClosed() {
32     }
33     @Override
34     public void onClick(int type, String adid) {
35         //type=1 加载视频 type=2 播放视频
36         if(type==1){
37             //加载视频 绑定激励视频回调
38             TestPlayVideo.getInstance()
39                 .load(AnimationFloatIconActivity.this,
40 bdAdvanceAnimationFloatIconAd);
41             }else if(type==2){
42                 //播放视频
43                 TestPlayVideo.getInstance()
44                     .play(AnimationFloatIconActivity.this);
45             }
46         }
47     }
```

```

44     });
45     //后台可配置是否打开关闭功能，打开时，点击关闭按钮触发该回调
46     bdAdvanceAnimationFloatIconAd.setBdAdvanceCloseViewListener(new
BDAdvanceCloseViewListener() {
47         @Override
48         public void onClose() {
49
50         }
51     });
52
53     //加载广告
54     bdAdvanceAnimationFloatIconAd.loadAd();
55
56     @Override
57     protected void onDestroy() {
58         super.onDestroy();
59         if (bdAdvanceAnimationFloatIconAd != null) {
60             bdAdvanceAnimationFloatIconAd.destroyAd();
61             bdAdvanceAnimationFloatIconAd = null;
62         }
63     }

```

4.6 Tab广告

Tab广告主要用于APP的Tab页面切换位置，直接展示在一级页面，开发者可直接调用Tab广告接口，获取广告view直接填充。

```

1 //activity 必须为activity
2 //adspotId 广告位ID
3 public void loadTabAd(){
4     BDAdvanceTabAd bxmTab = new BDAdvanceTabAd(activity, adspotId);
5     bxmTab.setBdAdvanceListener(new BDAdvanceTabListener() {
6         @Override
7         public void onAdLoad(BDAdvanceTabAdItem adItem) {
8             showTabAdItem(adItem);
9         }
10        @Override
11        public void onAdFailed(String msg) {
12        }
13    });
14    bxmTab.loadAd();
15 }
16
17 BDAdvanceTabAdItem tabAdItem;
18 public void showTabAdItem(final BDAdvanceTabAdItem tabAdItem) {
19     if(tabAdItem!=null){
20         tabAdItem.setAdListener(new BDAdvanceTabAdItem.TabAdListener()
21         {
22             @Override
23             public void onAdShow() {
24             }
25             @Override
26             public void onAdFailed() {
27             }
28             @Override
29             public void onClick(int type, String adid) {
30                 //type=1 加载视频 type=2 播放视频
31                 if(type==1){
32                     //加载视频 绑定激励视频回调
33                     TestPlayVideo.getInstance()
34                         .load(getActivity(), (BDAdvanceBaseAppNative)
35                         tabAdItem);
36                 }else if(type==2){
37                     //播放视频
38                     TestPlayVideo.getInstance()
39                         .play(getActivity());
40                 }
41             }
42         });
43         this.tabAdItem = tabAdItem;
44         ll_container.setVisibility(View.VISIBLE);
45         ll_container.removeAllViews();
46         ll_container.addView(tabAdItem.getView());
47         tabAdItem.render();
48     }
49 }

```

```
47     }
48     @Override
49     public void onDestroy() {
50         super.onDestroy();
51         if(tabAdItem!=null) {
52             tabAdItem.destroy();
53         }
54     }
```

4.7 聚合图文广告

聚合图文广告是非标广告的一种，聚合了多个图文类型广告，通过表格的展示形式渲染到容器中。

```
1      BDAdvanceMultiplePicTextAd bdAdvanceMultiplePicTextAd;
2
3      private void loadAd() {
4          // adContainer 展示广告容器, 宽高请按以下方式配置:
5          // android:layout_width="match_parent"
6          // android:layout_height="wrap_content"
7
8          // adspotId 广告位ID 变现猫后台申请
9          bdAdvanceMultiplePicTextAd = new BDAdvanceMultiplePicTextAd(
10             Context context, ViewGroup adContainer, String adspotId);
11          MultiplePicTextStyle style = new MultiplePicTextStyle.Builder()
12             .setPicTextStyle(new PicTextStyle.Builder()
13                 .setTextSize(12)
14                 .setTextColor(Color.parseColor("#333333"))
15                 .setPicTextSpacing(7)
16                 .build())
17             .setRowSpacing(15)
18             .setColumnSpacing(12)
19             .setChildAndContainerLeftSpacing(20)
20             .setChildAndContainerTopSpacing(15)
21             .setChildAndContainerRightSpacing(20)
22             .setChildAndContainerBottomSpacing(15)
23             .build();
24          bdAdvanceMultiplePicTextAd.setMultiplePicTextStyle(style);
25          bdAdvanceMultiplePicTextAd.setBdAdvanceMultiplePicTextListener(
26             new BDAdvanceMultiplePicTextListener() {
27                 @Override
28                 public void onAdShow() {
29                     // 广告展示
30                 }
31
32                 @Override
33                 public void onAdFailed() {
34                     // 广告加载失败
35                 }
36
37                 @Override
38                 public void onChildAdClicked(int position) {
39                     // 广告点击, position -> 子广告下标
40                 }
41
42                 @Override
43                 public void onChildAdClosed(int position) {
44                     // 广告关闭, position -> 子广告下标
45                 }
46
47                 @Override
48                 public void onChildActivityClosed(int position) {
```

```

49         // 活动页关闭, position -> 子广告下标
50     }
51
52     @Override
53     public void onChildAppNativeClick(int type, String adid, int
position) {
54         //type -> 1 加载视频、2 播放视频, position -> 子广告下标
55         if (type == 1) {
56             //加载视频 bdAdvanceMultiplePicTextAd 绑定激励视频回调
57             TestPlayVideo.getInstance()
58
59             .load(MultiplePicTextActivity.this, bdAdvanceMultiplePicTextAd);
60             } else if (type == 2) {
61                 //播放视频
62
63                 TestPlayVideo.getInstance().play(MultiplePicTextActivity.this);
64             }
65         });
66
67         //加载广告
68         bdAdvanceMultiplePicTextAd.loadAds();
69     }
70
71     @Override
72     protected void onDestroy() {
73         super.onDestroy();
74         if (bdAdvanceMultiplePicTextAd != null) {
75             bdAdvanceMultiplePicTextAd.destroyAd();
76             bdAdvanceMultiplePicTextAd = null;
77         }
78     }

```

自定义样式参数说明

MultiplePicTextStyle表

参数名	含义	默认值 (单位: dp)
setPicTextStyle()	设置子广告位样式	见PicTextStyle表详细说明
setRowSpacing()	行间距	15
setColumnSpacing()	列间距	12
setChildAndContainerSpacing()	图文广告和容器四边间距	-
setChildAndContainerLeftSpacing()	图文广告和容器左边距	20
setChildAndContainerTopSpacing()	图文广告和容器上边距	15
setChildAndContainerRightSpacing()	图文广告和容器右边距	20
setChildAndContainerBottomSpacing() ()	图文广告和容器下边距	15

PicTextStyle表

Tip: 以下属性配置会作用于所有子广告位上

参数名	含义	默认值
setTextSize()	设置子广告文字大小	12sp
setTextColor()	设置子广告文字颜色	Color.parseColor("#333333")
setPicWidth()	设置子广告图片宽度	自适应 最大宽度限制=(容器宽度-列间距-左右边距) / 单行子广告个数
setPicHeight()	设置子广告图片高度	等于图片宽度
setPicTextSpacing()	设置子广告图片和文字间距	7dp

注: 图片的ScaleType为FIT_CENTER, 这个属性会保证图片完整居中显示, 不拉伸。

4.8 聚合非标Banner广告

聚合非标Banner广告是非标广告的一种，聚合了多个非标Banner类型广告，通过表格的展示形式渲染到容器中。

```
1      BDAdvanceMultipleCustomBannerAd bdAdvanceMultipleCustomBannerAd;
2
3      private void loadAd() {
4          // adContainer 展示广告容器, 宽高请按以下方式配置:
5          // android:layout_width="match_parent"
6          // android:layout_height="wrap_content"
7
8          // adspotId 广告位ID 变现猫后台申请
9          bdAdvanceMultipleCustomBannerAd = new
BDAdvanceMultipleCustomBannerAd(
10              Context context, ViewGroup adContainer, String adspotId);
11          MultipleCustomBannerStyle style = new
MultipleCustomBannerStyle.Builder()
12              .setRowSpacing(10)
13              .setColumnSpacing(10)
14              .setChildAndContainerLeftSpacing(20)
15              .setChildAndContainerTopSpacing(15)
16              .setChildAndContainerRightSpacing(20)
17              .setChildAndContainerBottomSpacing(15)
18              .build();
19
20          bdAdvanceMultipleCustomBannerAd.setMultipleCustomBannerStyle(style);
21
22          bdAdvanceMultipleCustomBannerAd.setBdAdvanceMultipleCustomBannerListener(
23              new BDAdvanceMultipleCustomBannerListener() {
24                  @Override
25                  public void onAdShow() {
26                      // 广告展示
27                  }
28
29                  @Override
30                  public void onAdFailed() {
31                      // 广告加载失败
32                  }
33
34                  @Override
35                  public void onChildAdClicked(int position) {
36                      // 广告点击, position -> 子广告下标
37                  }
38
39                  @Override
40                  public void onChildAdClosed(int position) {
41                      // 广告关闭, position -> 子广告下标
42                  }
43
44                  @Override
45                  public void onChildActivityClosed(int position) {
46                      // 活动页关闭, position -> 子广告下标
```

```

45         }
46
47         @Override
48         public void onChildAppNativeClick(int type, String adid, int
position) {
49             //type -> 1 加载视频、2 播放视频, position -> 子广告下标
50             if (type == 1) {
51                 //加载视频 bdAdvanceMultipleCustomBannerAd 绑定激励视频回调
52                 TestPlayVideo.getInstance()
53                     .load(MultipleCustomBannerActivity.this
54                         , bdAdvanceMultipleCustomBannerAd);
55             } else if (type == 2) {
56                 //播放视频
57
58                 TestPlayVideo.getInstance().play(MultipleCustomBannerActivity.this);
59             }
60         });
61
62         //加载广告
63         bdAdvanceMultipleCustomBannerAd.loadAds();
64     }
65
66     @Override
67     protected void onDestroy() {
68         super.onDestroy();
69         if (bdAdvanceMultipleCustomBannerAd != null) {
70             bdAdvanceMultipleCustomBannerAd.destroyAd();
71             bdAdvanceMultipleCustomBannerAd = null;
72         }
73     }

```

自定义样式参数说明

MultipleCustomBannerStyle表

参数名	含义	默认值 (单位: dp)
setCustomBannerStyle()	设置子广告位样式	见CustomBannerStyle表 详细说明
setRowSpacing()	行间距	10
setColumnSpacing()	列间距	10
setChildAndContainerSpacing()	非标banner和容器四边间距	-
setChildAndContainerLeftSpacing()	非标banner和容器左边距	20
setChildAndContainerTopSpacing()	非标banner和容器上边距	15
setChildAndContainerRightSpacing()	非标banner和容器右边距	20
setChildAndContainerBottomSpacing() ()	非标banner和容器下边距	15

CustomBannerStyle表

Tip: 以下属性配置会作用于所有子广告位上

参数名	含义	默认值
setPicWidth()	设置子广告图片宽度	自适应 最大宽度限制=(容器宽度- 列间距-左右边距) / 单行 子广告个数
setPicHeight()	设置子广告图片高度	自适应

*4.9 接入示例：互动激励视频

变现猫为开发者搭建互动激励场景，活动内播放激励视频，获得广告收益

- 1、需要在变现猫ssp后台创建活动入口、激励视频两类广告位
- 2、开发者使用活动入口广告位id对应接口，调用入口广告位，获得入口素材及活动落地页
- 3、进入活动后，参与环节中会出激励视频弹窗，点击播放视频调用激励视频，激励视频播放完成用户获得对应奖励



第一步（入口广告位类型为非标banner接入示例）

活动入口请调用入口广告位id对应类型的接口，可在SSP后台查看。

如：浮标广告、非标banner广告、自渲染。

以下是：入口广告位类型为非标banner接入示例

```
1 //adContainer 展示广告容器
2 //adspotId 广告位ID 变现猫后台创建
3 bdAdvanceButtonAd = new BDAdvanceButtonAd(Activity activity, ViewGroup
  adContainer, String adspotId);
4 //2.设置回调
5 bdAdvanceButtonAd.setBdAdvanceButtonListener(new BDAdvanceButtonListener()
  {
6     @Override
7     public void onActivityClosed() {
8         //活动页关闭
9     }
10    @Override
11    public void onAdShow() {
12        //广告展示
13    }
14    @Override
15    public void onAdFailed() {
16        //广告加载失败
17    }
18    @Override
19    public void onAdClicked() {
20        //广告点击
21    }
22 });
23 //互动激励视频
24 bdAdvanceButtonAd.setBdAppNativeOnClickListener(new
  BDAppNativeOnClickListener() {
25     @Override
26     public void onActivityClosed() {
27
28     }
29
30     @Override
31     public void onClick(int type, String adid) {
32         //type=1 加载视频 type=2 播放视频
33         if(type==1){
34             //加载视频 bdAdvanceButtonAd 绑定激励视频回调
35             TestPlayVideo.getInstance()
36                 .load(ButtonAdActivity.this, bdAdvanceButtonAd);
37         }else if(type==2){
38             //播放视频
39             TestPlayVideo.getInstance()
40                 .play(ButtonAdActivity.this);
41         }
42     }
43 });
44 //3.加载广告
45 bdAdvanceButtonAd.loadAd();
```

第二步（播放穿山甲激励视频）

SDK互动激励视频回调说明

```
void onADLoad(); //视频加载完成回调，活动会在之后执行播放逻辑
void onReward(); //奖励回调，只有在收到onReward后，在onADClose时才会发奖
void onADClose(); //视频页面关闭回调，活动页面将会继续，请注意回调时机
void onError(int code); //出错回调，SDK会执行下一个渠道
void onSkipped(); //用户跳过
void onADShow(); //视频开始曝光
void onADClick(); //点击回调
void onVideoComplete(); //播放完成回调
```

注意：必须将以上视频播放状态回调给SDK

注意：必须将以上视频播放状态回调给SDK

注意：必须将以上视频播放状态回调给SDK

穿山甲激励视频接入示例

```
1 public class TestPlayVideo {
2     static TestPlayVideo instance = new TestPlayVideo();
3     public static TestPlayVideo getInstance(){
4         return instance;
5     }
6
7     String appid = "5060085";
8     String adspotid = "945132252";
9
10    //一、加载穿山甲激励视频
11    //BDAdvanceBaseAppNative 绑定激励视频回调
12    public void load(final Activity activity, final BDAdvanceBaseAppNative
appNative) {
13
14        try {
15            //初始化
16            TAdSdk.init(activity, new TAdConfig.Builder()
17                .appId(appid)
18                .useTextureView(true) //使用TextureView控件播放视频,默认为
SurfaceView,当有SurfaceView冲突的场景,可以使用TextureView
19                .appName("appName")
20                .titleBarTheme(TAdConstant.TITLE_BAR_THEME_DARK)
21                .allowShowNotify(true) //是否允许sdk展示通知栏提示
22                .allowShowPageWhenScreenLock(true) //是否在锁屏场景支持展
示广告落地页
23                .debug(false) //测试阶段打开,可以通过日志排查问题,上线时去除
该调用
24
25                .directDownloadNetworkType(TAdConstant.NETWORK_STATE_WIFI,
TAdConstant.NETWORK_STATE_4G) //允许直接下载的网络状态集合
26                .supportMultiProcess(true)//是否支持多进程
27                .needClearTaskReset()
28                .build(), new TAdSdk.InitCallback() {
29                    @Override
30                    public void success() {
31
32
33                    @Override
34                    public void fail(int i, String s) {
35
36
37                });
38
39                //创建TAdNative对象
40                TAdNative adNative =
TAdSdk.getAdManager().createAdNative(activity);
41
```



```

42         //模版渲染请求AdSlot
43         AdSlot adSlot = new AdSlot.Builder()
44             .setCodeId(adspotid)
45             .setSupportDeepLink(true)
46             .setAdCount(1)
47             .setRewardName("coin")
48             .setRewardAmount(1)
49             .setExpressViewAcceptedSize(500,500)//个性化模板广告需要
设置期望个性化模板广告的大小,单位dp,激励视频场景,只要设置的值大于0即可。仅模板广告需要
设置此参数
50             .setImageAcceptedSize(1080, 1920)
51             .setOrientation(TTAdConstant.VERTICAL)
52             .setMediaExtra("media_extra")
53             .setUserID("user123")
54             .build();
55
56         //加载激励视频广告
57         adNative.loadRewardVideoAd(adSlot, new
TTAdNative.RewardVideoAdListener() {
58             @Override
59             public void onError(int i, String s) {
60                 appNative.onError(i);
61             }
62
63             @Override
64             public void onRewardVideoAdLoad(TTRewardVideoAd
ttRewardVideoAd) {
65                 if(ttRewardVideoAd==null){
66                     appNative.onError(0);
67                 }else{
68                     bindRewardListener(ttRewardVideoAd, appNative);
69                     appNative.onADLoad();
70                 }
71             }
72
73             @Override
74             public void onRewardVideoCached() {
75
76             }
77
78             @Override
79             public void onRewardVideoCached(TTRewardVideoAd
ttRewardVideoAd) {
80
81             }
82         });
83     } catch (Throwable e) {
84         appNative.onError(0);
85     }
86 }

```

```

87
88
89 //二、绑定穿山甲激励视频回调
90 //BDAdvanceBaseAppNative 绑定激励视频回调
91 TTRewardVideoAd mttRewardVideoAd;
92 private void bindRewardListener(TTRewardVideoAd ttRewardVideoAd, final
BDAdvanceBaseAppNative appNative) {
93     mttRewardVideoAd = ttRewardVideoAd;
94     mttRewardVideoAd.setRewardAdInteractionListener(new
TTRewardVideoAd.RewardAdInteractionListener() {
95         @Override
96         public void onAdShow() {
97             appNative.onADShow();
98         }
99
100        @Override
101        public void onAdVideoBarClick() {
102            appNative.onADClick();
103        }
104
105        @Override
106        public void onAdClose() {
107            appNative.onADClose();
108        }
109
110        @Override
111        public void onVideoComplete() {
112            appNative.onVideoComplete();
113        }
114
115        @Override
116        public void onVideoError() {
117            appNative.onError(0);
118        }
119
120        @Override
121        public void onRewardVerify(boolean b, int i, String s, int i1,
String s1) {
122            appNative.onReward();
123        }
124
125        @Override
126        public void onSkippedVideo() {
127            appNative.onSkipped();
128        }
129    });
130
131 }
132
133 //三、播放穿山甲激励视频

```

```
134     public void play(Activity activity) {
135         if(mttRewardVideoAd!=null){
136             mttRewardVideoAd.showRewardVideoAd(activity);
137         }
138     }
139 }
140
```

5、标准广告位

5.1 开屏广告

开屏广告为用户在进入App时展示的全屏广告，以App启动作为曝光时机，提供5s左右的可感知广告展示。用户可以点击广告跳转到目标页面；或者点击右上角的“跳过”按钮，跳转到app内容首页。

```
1 //1.创建广告
2 //adspotId 广告位ID, 变现猫后台创建
3 //adContainer 展示广告父容器
4 //skipView 跳过view
5 //skipText 跳过字符串
6 advanceSplashAd = new BDAdvanceSplashAd(Activity activity,
7     String adspotId, ViewGroup adContainer, TextView skipView,
8     String skipText);
9 //2.设置图片素材尺寸和回调
10 advanceSplashAd.setCsjAcceptedSize(1080, 1920)//设置广告图片偏好尺寸
11 advanceSplashAd.setSplashListener(new BDAdvanceSplashListener() {
12     @Override
13     public void onClose() {
14         //广告关闭
15         jumpToMainActivity();
16     }
17     @Override
18     public void onAdSuccess() {
19         //广告加载成功
20     }
21     @Override
22     public void onAdShow() {
23         //展示成功
24     }
25     @Override
26     public void onAdFailed() {
27         //广告加载失败
28     }
29     @Override
30     public void onAdClicked() {
31         //广告点击
32     }
33 });
34 advanceSplashAd.setAppChannelListener(new BDAppChannelListener() {
35     @Override
36     public void onAppChannelAd() {
37         //处理本地广告
38         tvSkip.setVisibility(View.GONE);
39     }
40 });
41
42 TestCsjSplashAd.getInstance().loadAd(DispatchSplashAdActivity.this,
43     splashContainer, advanceSplashAd);
44
45 if (Build.VERSION.SDK_INT >= 23) {
46     checkAndRequestPermission();
47 } else {
```

```
45     //3.加载广告
46     advanceSplashAd.loadAD();
47 }
```

注：当开屏Activity处于后台时不要跳转主页面，设置一个变量来控制当前开屏页面是否可以跳转

Java | [复制代码](#)

```
1     /**
2     * 当开屏Activity处于后台时禁止跳转主页面
3     * 设置一个变量来控制当前开屏页面是否可以跳转
4     */
5     boolean canJump = false;
6     private void jumpToMainActivity() {
7         if (canJump) {
8             //跳转到主页面
9             startActivity(new Intent(this, MainActivity.class));
10        } else {
11            canJump = true;
12        }
13    }
14
15    @Override
16    protected void onPause() {
17        super.onPause();
18        canJump = false;
19    }
20
21    @Override
22    protected void onResume() {
23        super.onResume();
24        if (canJump) {
25            jumpToMainActivity();
26        }
27        canJump = true;
28    }
```

5.2 信息流

信息流支持图文和视频，提供了多种的广告模板样式，开发者可直接调用信息流广告接口，获取对应模板的广告view直接填充，开发者可以随时对平台上勾选的模板进行样式上的调整。可点击关闭按钮关闭。

```

1  public class DispatchNativeExpressAdActivity extends Activity
   implements View.OnClickListener {
2      private FrameLayout flContainer;
3      @Override
4      protected void onCreate(@Nullable Bundle savedInstanceState) {
5          super.onCreate(savedInstanceState);
6          setContentView(R.layout.activity_dispatch_express_ad);
7          flContainer = findViewById(R.id.fl_native_express_container);
8          findViewById(R.id.bt_load_express_ad).setOnClickListener(this);
9          BDManager.getStance().requestPermission(this);
10     }
11     @Override
12     public void onClick(View view) {
13         if (view.getId() == R.id.bt_load_express_ad) {
14             loadAd();
15         }
16     }
17
18     private void loadAd() {
19         //1.创建信息流广告
20         //adspotId 广告位ID 变现猫后台创建
21         BDAdvanceNativeExpressAd bdAdvanceNativeExpressAd
22 = new BDAdvanceNativeExpressAd(Activity activity, String adspotId);
23         bdAdvanceNativeExpressAd.setAdCount(3) //可以设置最大请求条数, 建议1~3
   条
24         //2.设置回调
25         bdAdvanceNativeExpressAd.registerBxmAdvanceExpressListener(new BDA
   dvanceNativeExpressListener() {
26             @Override
27             public void onLoadExpressList(List<BDAdvanceNativeExpressAdIte
   m> list) {
28                 //广告数据加载成功
29                 BDAdvanceNativeExpressAdItem bdAdvanceNativeExpressAdItem
   = list.get(0);
30                 //4.调用渲染
31                 bdAdvanceNativeExpressAdItem.render();
32             }
33             @Override
34             public void onAdClose(View view) {
35                 //广告关闭
36             }
37             @Override
38             public void onAdRenderFailed(View view) {
39                 //广告渲染失败
40             }
41             @Override
42             public void onAdRenderSuccess(View view, float width,
   float height) {

```

```

42         //广告渲染成功
43         flContainer.removeAllViews();
44         flContainer.setVisibility(View.VISIBLE);
45         flContainer.addView(view);
46     }
47     @Override
48     public void onAdShow() {
49         //广告展示
50     }
51     @Override
52     public void onAdFailed() {
53         //广告加载失败
54     }
55     @Override
56     public void onAdClicked() {
57         //广告点击
58     }
59 });
60     bdAdvanceNativeExpressAd.setAppChannelListener(new
BDAppChannelListener() {
61         @Override
62         public void onAppChannelAd() {
63             //处理本地广告
TestCsjNativeExpressAd.getInstance().loadAd(DispatchNativeExpressAdActivit
y.this, flContainer, bdAdvanceNativeExpressAd);
64         }
65     });
66     //3.加载广告
67     bdAdvanceNativeExpressAd.loadAD();
68 }
69 }

```

5.3 Banner广告

Banner广告(横幅广告)位于app顶部、中部、底部任意一处，展现场景非常灵活。可点击关闭按钮关闭。

```
1 public class DispatchBannerAdActivity extends Activity{
2     private ViewGroup container;
3     private BDAdvanceBannerAd bdAdvanceBannerAd;
4     @Override
5     protected void onCreate(@Nullable Bundle savedInstanceState) {
6         super.onCreate(savedInstanceState);
7         setContentView(R.layout.activity_dispatch_banner_ad);
8         container = findViewById(R.id.fl_ad_container);
9         BDManager.getStance().requestPermission(this);
10        initBannerAd();
11    }
12    private void initBannerAd() {
13        //1.创建banner广告
14        //adContainer 展示广告父容器
15        //adspotId 广告位ID, 变现猫后台创建
16        bdAdvanceBannerAd = new BDAdvanceBannerAd(Activity activity,
17        ViewGroup adContainer, String adspotId);
18        //2.设置回调
19        bdAdvanceBannerAd.setBDAdvanceBannerListener(new BDAdvanceBannerLi
20        stener() {
21            //点击了关闭按钮, 可以移除横幅广告
22            @Override
23            public void onDislike() {
24            }
25            //广告展示
26            @Override
27            public void onAdShow() {
28            }
29            //广告加载失败
30            @Override
31            public void onAdFailed() {
32            }
33            //广告被点击
34            @Override
35            public void onAdClicked() {
36            }
37        });
38        bdAdvanceBannerAd.setAppChannelListener(new BDAppChannelListener()
39        {
40            @Override
41            public void onAppChannelAd() {
42                //处理本地广告
43            }
44        });
45        TestCsjBannerAd.getInstance().loadAd(DispatchBannerAdActivity.this,
46        container, bdAdvanceBannerAd);
47    }
48    }
49    };
```



```
44         //3.加载广告
45         bdAdvanceBannerAd.loadAD();
46     }
47
48     @Override
49     protected void onDestroy() {
50         super.onDestroy();
51         //4.activity销毁后调用destroy
52         if (bdAdvanceBannerAd != null)bdAdvanceBannerAd.destroy();
53     }
54 }
```

5.4 插屏广告

插屏广告是一种非全屏广告形式，适合在应用执行流程的自然停顿点展示，当应用展示插屏广告时，用户可以选择点按广告，也可以将其关闭，返回应用。

```
1 public class DispatchInteractionActivity extends Activity
  implements View.OnClickListener {
2     private BDAdvanceInteractionAd bdAdvanceInteractionAd;
3     @Override
4     protected void onCreate(@Nullable Bundle savedInstanceState) {
5         super.onCreate(savedInstanceState);
6         setContentView(R.layout.activity_dispatch_interaction);
7         findViewById(R.id.bt_load_interaction_ad).setOnClickListener(this)
8     ;
9         BDManager.getStance().requestPermission(this);
10    }
11    @Override
12    public void onClick(View view) {
13        if (view.getId() == R.id.bt_load_interaction_ad) {
14            loadAd();
15        } else if (view.getId() == R.id.bt_show_interaction_ad) {
16            if (isReady){
17                bdAdvanceInteractionAd.showAd();
18            }else if(isReadyAppChannel){
19
20                TestCsjInteractionAd.getInstance().showInteractionAd(DispatchInteractionA
21                ctivity.this);
22            }
23        }
24    }
25    private void loadAd() {
26        //1.创建插屏广告
27        //adspotId 广告位ID 变现猫后台创建
28        bdAdvanceInteractionAd = new BDAdvanceInteractionAd(Activity
29        activity, String adspotId);
30        //2.设置回调
31        bdAdvanceInteractionAd.setBdAdvanceInteractionListener(new BDAdvan
32        ceInteractionListener() {
33            //广告关闭
34            @Override
35            public void onAdClose() {
36            }
37            //广告加载成功;可进行展示操作
38            @Override
39            public void onAdSuccess() {
40                bdAdvanceInteractionAd.showAd();
41            }
42            //广告展示
43            @Override
44            public void onAdShow() {
45            }
46            //广告加载失败
```

```

43         @Override
44         public void onAdFailed() {
45         }
46         //广告被点击
47         @Override
48         public void onAdClicked() {
49         }
50     });
51     bdAdvanceInteractionAd.setAppChannelListener(new
BDAppChannelListener() {
52         @Override
53         public void onAppChannelAd() {
54             //处理本地广告
TestCsjInteractionAd.getInstance().loadAd(DispatchInteractionActivity.this
, bdAdvanceInteractionAd);
55             isReadyAppChannel = true;
56         }
57     });
58     //3.加载广告
59     bdAdvanceInteractionAd.loadAD();
60 }
61
62 @Override
63 protected void onDestroy() {
64     super.onDestroy();
65     //4.activity销毁后调用destroy
66     if (bdAdvanceInteractionAd != null) {
67         bdAdvanceInteractionAd.destroy();
68     }
69 }
70 }

```

5.5 激励视频

激励视频广告是指将短视频融入到app场景当中，用户观看短视频广告后可以得到一些应用内奖励。

```
1 public class DispatchRewardVideoActivity extends Activity
  implements View.OnClickListener {
2     private BDAdvanceRewardAd bdAdvanceRewardAd;
3     private boolean isReady;
4     @Override
5     protected void onCreate(@Nullable Bundle savedInstanceState) {
6         super.onCreate(savedInstanceState);
7         setContentView(R.layout.demo_reward_video_ad);
8         findViewById(R.id.bt_load_reward_ad).setOnClickListener(this);
9         findViewById(R.id.bt_show_reward_ad).setOnClickListener(this);
10
11         BDManager.getStance().requestPermission(this);
12     }
13     @Override
14     public void onClick(View view) {
15         switch (view.getId()) {
16             case R.id.bt_load_reward_ad:
17                 loadAd();
18                 break;
19             case R.id.bt_show_reward_ad:
20                 showAd();
21                 break;
22         }
23     }
24     private void loadAd() {
25         //1.创建激励视频广告
26         //adspotId 广告位ID 变现猫后台创建
27         bdAdvanceRewardAd = new BDAdvanceRewardAd(Activity activity,
  String adspotId);
28         //2.设置回调
29         bdAdvanceRewardAd.setBdAdvanceRewardListener(new BDAdvanceRewardLi
  stener() {
30             //广告加载成功,可进行播放操作
31             @Override
32             public void onAdLoad() {
33                 isReady = true;
34             }
35             //播放完成
36             @Override
37             public void onPlayCompleted() {
38             }
39             //广告关闭
40             @Override
41             public void onAdClose() {
42             }
43             //激励视频广告播放达到激励条件回调
44             @Override
```

```

45         public void onReward() {
46         }
47         //广告展示
48         @Override
49         public void onAdShow() {
50         }
51         //广告加载失败
52         @Override
53         public void onAdFailed() {
54             isReady = false;
55         }
56         //广告被点击
57         @Override
58         public void onAdClicked() {
59         }
60     });
61     //3.加载广告
62     bdAdvanceRewardAd.loadAD();
63 }
64
65 private void showAd() {
66     //4.视频加载成功后才可进行播放
67     if (isReady) {
68         bdAdvanceRewardAd.showAd();
69     }
70 }
71 }

```

5.6 沉浸式视频

沉浸式视频适合在竖版全屏视频流中使用，类似于短视频大屏模式。

```

1  public class DispatchFeedVideoActivity extends Activity implements
View.OnClickListener {
2      FrameLayout fl_container;
3      BDAdvanceFeedVideoAd mADManager;
4      BDAdvanceFeedVideoAdItem mAdItem;
5      @Override
6      protected void onCreate(@Nullable Bundle savedInstanceState) {
7          super.onCreate(savedInstanceState);
8          setContentView(R.layout.activity_dispatch_feed_video);
9          fl_container = findViewById(R.id.fl_container);
10         findViewById(R.id.bt_load_reward_ad).setOnClickListener(this);
11         findViewById(R.id.bt_show_reward_ad).setOnClickListener(this);
12
13         findViewById(R.id.bt_show_reward_ad_destroy).setOnClickListener(this);
14         BDManager.getStance().requestPermission(this);
15     }
16     @Override
17     public void onClick(View view) {
18         int viewId = view.getId();
19         switch (viewId) {
20             case R.id.bt_load_reward_ad:
21                 loadAd();
22                 break;
23             case R.id.bt_show_reward_ad:
24                 showAd();
25                 break;
26             case R.id.bt_show_reward_ad_destroy:
27                 destroyAd();
28                 break;
29         }
30     }
31     /**
32      * 初始化加载广告
33      */
34     private void loadAd() {
35         mADManager = new BDAdvanceFeedVideoAd(this,
36         Constants.FEED_VIDEO_AD_SPOT_ID);
37         mADManager.setBDAdvanceFeedVideoListener(new
38         BDAdvanceFeedVideoListener() {
39             @Override
40             public void onLoadList(List<BDAdvanceFeedVideoAdItem>
41             mAdItemList) {
42                 if(mAdItemList==null || mAdItemList.size()==0){
43                     return;
44                 }
45                 mAdItem = mAdItemList.get(0);
46                 mAdItem.registerViewForInteraction(new
47                 BDAdvanceFeedVideoListener.FeedVideoAdListener() {

```

```

43         @Override
44         public void onPlayCompleted() {
45             Toast.makeText(DispatchFeedVideoActivity.this, "播
放完成", Toast.LENGTH_SHORT).show();
46         }
47         @Override
48         public void onPlayError() {
49             Toast.makeText(DispatchFeedVideoActivity.this, "播
放错误", Toast.LENGTH_SHORT).show();
50         }
51         @Override
52         public void onAdClicked() {
53             Toast.makeText(DispatchFeedVideoActivity.this, "广
告点击", Toast.LENGTH_SHORT).show();
54         }
55         @Override
56         public void onAdShow() {
57             Toast.makeText(DispatchFeedVideoActivity.this, "广
告展示", Toast.LENGTH_SHORT).show();
58         }
59         @Override
60         public void onAdClose() {
61             Toast.makeText(DispatchFeedVideoActivity.this, "广
告关闭", Toast.LENGTH_SHORT).show();
62         }
63     });
64     /**
65     * 预加载视频
66     */
67     mAdItem.preloading(new
BDAdvanceFeedVideoListener.FeedVideoPreloadingListener() {
68         @Override
69         public void onStart() {
70             Toast.makeText(DispatchFeedVideoActivity.this, "预
加载onStart", Toast.LENGTH_SHORT).show();
71         }
72
73         @Override
74         public void onFinish() {
75             Toast.makeText(DispatchFeedVideoActivity.this, "预
加载onFinish", Toast.LENGTH_SHORT).show();
76         }
77
78         @Override
79         public void onFailure(String msg) {
80             Toast.makeText(DispatchFeedVideoActivity.this, "预
加载onFailure", Toast.LENGTH_SHORT).show();
81         }
82     });
83 }

```

```

84         @Override
85         public void onAdFailed() {
86             Toast.makeText(DispatchFeedVideoActivity.this, "广告加载失
败", Toast.LENGTH_SHORT).show();
87         }
88     });
89     /**
90      * 加载广告
91      */
92     mADManager.loadAD();
93 }
94 /**
95  * 展示广告
96  */
97 private void showAd() {
98     if (mAdItem!=null) {
99         fl_container.removeAllViews();
100        fl_container.addView(mAdItem.getAdView());
101        mAdItem.render();
102    }
103 }
104 /**
105  * 销毁广告
106  */
107 private void destroyAd(){
108     if(mAdItem!=null){
109         mAdItem.destroy();
110     }
111 }
112 @Override
113 protected void onDestroy() {
114     super.onDestroy();
115     destroyAd();
116 }
117 }

```

5.7 全屏视频

全屏视频广告，该广告的效果播放全屏的视频。


```
1 public class DispatchFullVideoActivity extends Activity
  implements View.OnClickListener {
2     private BDAdvanceFullVideoAd bdAdvanceFullVideoAd;
3     private boolean isReady = false;
4     @Override
5     protected void onCreate(@Nullable Bundle savedInstanceState) {
6         super.onCreate(savedInstanceState);
7         setContentView(R.layout.activity_dispatch_full_video);
8         findViewById(R.id.bt_load_full_video_ad).setOnClickListener(this);
9         findViewById(R.id.bt_show_full_video_ad).setOnClickListener(this);
10
11         BDManager.getStance().requestPermission(this);
12     }
13     @Override
14     public void onClick(View view) {
15         switch (view.getId()) {
16             case R.id.bt_load_full_video_ad:
17                 loadAd();
18                 break;
19             case R.id.bt_show_full_video_ad:
20                 showAd();
21                 break;
22         }
23     }
24     private void loadAd() {
25         //1.创建全屏视频广告
26         //adspotId 广告位ID 变现猫后台创建
27         bdAdvanceFullVideoAd = new BDAdvanceFullVideoAd(Activity activity,
  String adspotId);
28         //2.设置回调
29         bdAdvanceFullVideoAd.setBdAdvanceFullVideoListener(new BDAdvanceFu
  llVideoListener() {
30             //广告加载成功,可进行播放操作
31             @Override
32             public void onAdLoad() {
33                 isReady = true;
34             }
35             //广告展示
36             @Override
37             public void onAdShow() {
38
39             }
40             //广告关闭
41             @Override
42             public void onAdClose() {
43                 }
44             //播放完成
```

```

45         @Override
46         public void onVideoComplete() {
47             }
48             //跳过视频回调
49         @Override
50         public void onSkippedVideo() {
51             }
52             //广告加载失败
53         @Override
54         public void onAdFailed() {
55             isReady = false ;
56         }
57             //广告被点击
58         @Override
59         public void onAdClicked() {
60             }
61     });
62     //3.加载广告
63     bdAdvanceFullVideoAd.loadAD();
64 }
65
66     private void showAd() {
67         //4.视频加载成功后才可进行播放
68         if (isReady) {
69             bdAdvanceFullVideoAd.showAd();
70         }
71     }
72 }

```

*5.8 接入示例：本地广告

本地广告是基于 SDK下发本地渠道，由媒体 APP内自行实现本地的广告效果，媒体对接过程中，需要遵循本地调用逻辑跟SDK进行交互绑定，否则将影响广告效果。

目前支持本地广告模块有：开屏，插屏，Banner，信息流模板

注：设置本地监听方法setAppChannelListener，如果没有本地广告，请不要设置。

第一步（入口广告位类型为开屏接入示例）

```
1 //1.创建广告
2 //adspotId 广告位ID, 变现猫后台创建
3 //adContainer 展示广告父容器
4 //skipView 跳过view
5 //skipText 跳过字符串
6 advanceSplashAd = new BDAdvanceSplashAd(Activity activity,
7     String adspotId, ViewGroup adContainer, TextView skipView,
8     String skipText);
9 //2.设置图片素材尺寸和回调
10 advanceSplashAd.setCsjAcceptedSize(1080, 1920)//设置广告图片偏好尺寸
11 advanceSplashAd.setSplashListener(new BDAdvanceSplashListener() {
12     @Override
13     public void onClose() {
14         //广告关闭
15     }
16     @Override
17     public void onAdSuccess() {
18         //广告加载成功
19     }
20     @Override
21     public void onAdShow() {
22         //展示成功
23     }
24     @Override
25     public void onAdFailed() {
26         //广告加载失败
27     }
28     @Override
29     public void onAdClicked() {
30         //广告点击
31     }
32 });
33 //设置本地监听, 如果没有本地广告, 请不要设置
34 advanceSplashAd.setAppChannelListener(new BDAppChannelListener() {
35     @Override
36     public void onAppChannelAd() {
37         //处理本地广告
38         tvSkip.setVisibility(View.GONE);
39     }
40 });
41
42 if (Build.VERSION.SDK_INT >= 23) {
43     checkAndRequestPermission();
44 } else {
```

```
45     //3.加载广告
46     advanceSplashAd.loadAD();
47 }
```

第二步（穿山甲开屏广告）

本地回调说明

```
void onAppAdSuccess(); //加载成功
void onAppAdFailed(int code, String msg); //错误回调，SDK会执行下一个渠道
void onAppAdShow(); //曝光
void onAppAdClicked(); //点击
void onAppAdClose(); //关闭
```

注意：必须将以上结果回调给SDK

注意：必须将以上结果回调给SDK

注意：必须将以上结果回调给SDK

穿山甲开屏接入示例

```
1 public class TestCsjSplashAd {
2     static TestCsjSplashAd instance = new TestCsjSplashAd();
3     public static TestCsjSplashAd getInstance(){
4         return instance;
5     }
6
7     String appid = "5060085";
8     String adspotid = "887315306";
9
10    public void loadAd(final Activity context, final ViewGroup
adContainer, final BDAvanceSplashAd splashAdListener) {
11        try {
12            TestCsjManagerHolder.init(context, appid);
13            TAdManager ttAdManager = TestCsjManagerHolder.get();
14            TAdNative adNative = ttAdManager.createAdNative(context);
15            AdSlot adSlot = (new AdSlot.Builder())
16                .setCodeId(adspotid)
17                .setSupportDeepLink(true)
18                .setImageAcceptedSize(1080, 1920)
19                .build();
20            adNative.loadSplashAd(adSlot, new
TTAdNative.SplashAdListener() {
21                @Override
22                public void onError(int i, String s) {
23                    if (null != splashAdListener) {
24                        splashAdListener.onAppAdFailed(i, s);
25                    }
26                }
27
28                @Override
29                public void onTimeout() {
30                    if (null != splashAdListener) {
31
32                        splashAdListener.onAppAdFailed(Constant.ERROR_TIME_OUT, "");
33                    }
34                }
35
36                @Override
37                public void onSplashAdLoad(TTSplashAd ttSplashAd) {
38                    if (null == ttSplashAd) {
39                        if (null != splashAdListener) {
40
41                            splashAdListener.onAppAdFailed(Constant.ERROR_NO_DATA, "");
42                        }
43                    } else {
44                        View splashView = ttSplashAd.getSplashView();
45                        if (adContainer != null) {
46                            if (null != splashAdListener) {
```

```

45         splashAdListener.onAppAdSuccess();
46     }
47     adContainer.removeAllViews();
48     adContainer.setVisibility(View.VISIBLE);
49     adContainer.addView(splashView);
50 }
51
52     ttSplashAd.setSplashInteractionListener(new
TTSplashAd.AdInteractionListener() {
53         @Override
54         public void onAdClicked(View view, int i) {
55             if (null != splashAdListener) {
56                 splashAdListener.onAppAdClicked();
57             }
58         }
59
60         @Override
61         public void onAdShow(View view, int i) {
62             if (null != splashAdListener) {
63                 splashAdListener.onAppAdShow();
64             }
65         }
66
67         @Override
68         public void onAdSkip() {
69             context.finish();
70             if (null != splashAdListener) {
71                 splashAdListener.onAppAdClose();
72             }
73         }
74
75         @Override
76         public void onAdTimeOver() {
77             context.finish();
78             if (null != splashAdListener) {
79                 splashAdListener.onAppAdClose();
80             }
81         }
82     });
83 }
84 }
85     }, 5);
86 } catch (Throwable var3) {
87     if (null != splashAdListener) {
88         splashAdListener.onAppAdFailed(Constant.ERROR_EXCEPTION,
""");
89     }
90 }
91 }
92 }

```

6. 绑定用户数据（可选）

此接口将app的用户ID传入SDK，便于用户权益的打通，例如app可以通过变现猫api查询用户视频播放次数等，如有需要，app在获取到用户ID后调用此接口。

如果你选择对接该功能，请告知变现猫对接人，将你的应用配置为白名单才能生效。

Java | [复制代码](#)

```
1 //参数1.Context
2 //参数2.应用appid, 变现猫媒体后台获取
3 //参数3.用户id, 格式cuid=1122, 如果要加自定义参数,用&分开, cuid=1122&ext1=999,
4 //app可以在获取到用户id后调用此接口
5 BDManager.getStance().bindUserData(this,
  "e4524c9e54e943d1adcda6c0bd711e8a", "cuid=1122");
```

7. 常见问题排查

7.1 日志过滤

- LogTag: `bxm_sdk`

7.2 Log提示: "广告位ID非法"

- 请核对appid、广告位id、包名，三者是否与后台注册的一致。

7.3 Log提示: "无广告"

- 广告位id使用错误: 请核对后台注册的广告位场景类型是否与SDK接口类型一致。
- 广告位id没有配置广告: 请与运营同学确认广告位配置情况
- `enableAudit(false)`: 此接口用于打审核包专用,上线必须false, 设置为true则不出广点通等广告

7.4 Log提示: "please use GDTSDK in BXMSDK"

- 没有导入广点通SDK的aar:
请添加GDTSDK引用

Groovy | [复制代码](#)

```
1 implementation(name: 'GDTSDK.unionNormal.4.422.1292', ext: 'aar')
```

- 广点通SDK版本不匹配: **最低支持版本为 4.370.1240 ;**

7.5 Log提示: "BxmSdk is not init, please check."

- BxmSdk 没有初始化:
请调用

Java | [复制代码](#)

```
1 BDManager.getStance().init(this, "appid");
```

7.7 Log提示: “本地播放视频回调为空”

活动页面, 点击播放激励视频按钮, 没有播放, Log提示: “本地播放视频回调为空”
APP端需要处理本地视频播放, 可参考章节: [按钮接入示例](#), [互动激励视频](#)


```

1 //互动激励视频
2 bdAdvanceButtonAd.setBdAppNativeOnClickListener(new
  BAppNativeOnClickListener() {
3     @Override
4     public void onActivityClosed() {
5     }
6     @Override
7     public void onClick(int type, String adid) {
8         //type=1 加载视频    type=2 播放视频
9         if(type==1){
10            //加载视频 bdAdvanceFloatIconAd 绑定激励视频回调
11            TestPlayVideo.getInstance()
12                .load(ButtonAdActivity.this, bdAdvanceButtonAd);
13        }else if(type==2){
14            //播放视频
15            TestPlayVideo.getInstance()
16                .play(ButtonAdActivity.this);
17        }
18    }
19 });

```

7.8 活动页面，点击播放激励视频按钮，Toast提示：视频正在赶来，请重试

- 激励视频加载失败
- 请联系运营同学确认广告位配置的广告渠道，如：广点通、快手等
- 检查SSP后台的广告渠道配置信息是否正确，appid、广告位id、包名等
- 检查是否导入了对应广告渠道的SDK
- 检查是否处理了本地播放视频回调逻辑

修订历史

版本号	修订说明	修订时间
v3.5.6	1-新增 聚合图文、聚合非标banner 广告 2-穿山甲支持通过适配包动态集成	2021.11.05
v3.5.5.10	1-适配广点通4.422.1292版本、快手3.3.17版本	2021.11.04
v3.5.5	1-增加权限开关(默认打开): imei、定位、AppList 2-移除oaid获取逻辑，增加setOaid方法	2021.09.15

v3.5.4	1-开屏模块优化 2-增加错误统计功能	2021.08.18
v3.5.2	1-SDK优化	2021.07.15
v3.5.1	1-开屏、插屏、信息流、Banner 支持本地客户端 setAppChannelListener 2-激励视频发奖逻辑调整 3-自渲染回调事件调整	2021.07.08
v3.5.0.1	1-已知问题修复	2021.06.25
v3.5.0	1-新增活动预加载功能	2021.05.27
v3.4.7	1-快手版本更新3.3.9 2-SDK优化	2021.04.29
v3.4.6	1- 动画浮标：支持定时刷新和点击刷新 2-广点通版本更新4.333.1203 3- SDK优化	2021.04.13
v3.4.5	新增爱豆开屏、banner、信息流	2021.03.26
v3.4.4	新增动画浮标类型的广告位	2021.03.22
v3.4.2	1-SDK优化 2-oaid版本修改为1.0.13	2021.03.03
v3.4.0	1-下载类广告优化，（新增WgsFileProvider，请在AndroidManifest配置，详情可参考”步骤二“或示例工程） 2-SDK优化	2021.02.04
v3.3.2	1-SDK优化	2021.01.12
v3.3.1	1-增加百度开屏 2-增加快手开屏、激励视频、沉浸式视频	2020.12.30
v3.2.5	1-浮标广告：支持定时刷新、增加destroyAd()方法 2-广告类型丰富：活动出券支持非表单类的广告	2020.12.18
v3.2.4	1-调整穿山甲接入流程：由APP端自己实现穿山甲相关逻辑，SDK内移除所有穿山甲相关依赖。 2-重构并统一sdk相关名称：包名、类名、资源名等，APP接入只需重新import相关类路径。 3-丰富广告接入方式：新增Tab广告类型。	2020.12.11

	4-调整自渲染广告相关参数。 5-调整代码混淆部分（具体可参考接入文档或示例代码）	