

百度 Mobile SSP

移动应用推广 SDK

用户手册 (Android 版)

百度在线网络技术 (北京) 有限公司

(版权所有,翻版必究)

目录

一、	简介	4
二、	运行环境	4
三、	代码混淆	4
四、	SDK 包导入及权限配置	4
1.	申请应用的 APPSID	4
2.	导入 JAR 包	4
3.	配置 AndroidManifest.xml	5
4.	设置广告着陆页的颜色主题	6
五、	横幅 Banner 广告	6
六、	插屏广告	7
七、	开屏广告	8
八、	信息流广告	9
1.	ListView 控件中展示信息流广告	9
2.	普通广告区显示信息流广告（即 Native 原生广告）	11
3.	信息流视频	12
4.	信息流轮播模板	17
5.	信息流橱窗模板	19
九、	贴片广告	19
1.	前贴片广告	19
十、	内容联盟	21
十一、	联系我们	23
十二、	注意事项	23
	附件 1：申请 APP ID	23

一、简介

百度移动推广 SDK(Android)是百度官方推出的移动推广 SDK 在 Android 平台上的版本(以下简称 SDK)。SDK 的发行版本（完整下载包为 BaiduMobAds_MSSP_bd_SDK_android_v4.6.zip）中包括 JAR 包、javadoc 文档、Demo 示例、错误码对照表以及您正在阅读的用户手册。以下使用<SDK_PATH>代表 Baidu_MobAds_SDK.zip 解压后的根目录。

- JAR 包：<SDK_PATH>/Baidu_MobAds_SDK.jar，开发时导入工程中；
- 文档：<SDK_PATH>/Doc，提供 javadoc 帮助信息；
- 示例：<SDK_PATH>/BaiduSDKDemo，多个示例程序代码，帮助您迅速熟悉 SDK 的使用方式；
- 错误码对照表：<SDK_PATH>/Baidu_Mobads_sdk_4 错误码对照表，提供错误自查；
- 用户手册：本文档。

注：以下描述皆用 MSSP 表示 Mobile SSP。

二、运行环境

可运行于 **Android 2.2 (API Level 8) 及以上版本**。

三、代码混淆

如果您需要使用proguard混淆代码，需确保不要混淆SDK的代码。
请在proguard.cfg文件(或其他混淆文件)尾部添加如下配置：

```
-keep class com.baidu.** {  
    public protected *;  
}
```

注意：SDK代码被混淆后会导致广告无法展现或者其它异常

四、SDK 包导入及权限配置

1. 申请应用的 APPSID

登录百度 MSSP 网站后进行申请，您获取的 APP ID 适用于文档中的“BaiduMobAd_APP_ID”，详见 [附件 1](#)。

2. 导入 JAR 包

将百度推广的 jar 包复制到您工程的 Libs 目录下，并按如下步骤将其导入：

- 右键点击你的工程，然后选择“Properties”；
- 在工程属性对话框左边选择“Java Build Path”；
- 在工程属性对话框主窗口选择“Libraries”；
- 点击“Add JARs”，添加百度推广 jar 包。

3. 配置 AndroidManifest.xml

■ 添加 APPID

下文中的 `BaiduMobAd_APP_ID` 内容填写为您的：APP ID。

```
<meta-data android:name="BaiduMobAd_APP_ID" android:value="e866cfb0" />
```

或使用代码设置 `AdView.setAppSid(context, "e866cfb0");` // 第二个参数需改为您的 APPID
代码设置与配置文件设置是等价的，您只需要在一个地方设置即可。

特注：(1) `meta-data` 标签一定要在 `application` 标签下；(2) 将 APPSID 设置成 MSSP 网站上获取的 APP ID，即可广告调试，并且开始为您的 APP 计费；(3) 只有您提交的应用通过审核后，才会获得有效的 APP ID。

■ 添加 AppActivity 申明

```
<activity android:name="com.baidu.mobads.AppActivity"  
    android:configChanges="keyboard|keyboardHidden|orientation"  
    android:theme="@android:style/Theme.Translucent.NoTitleBar" />
```

■ 添加权限

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

另外，一些额外的权限可以让您应用获得更精准的广告投放，一份完整的配置范例如下：

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.READ_PHONE_STATE" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />  
<!-- 以下为可选权限 -->  
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />  
<uses-permission android:name="android.permission.RECORD_AUDIO" /> <uses-permission  
android:name="android.permission.VIBRATE" />  
<uses-permission android:name="android.permission.CAMERA" />  
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

4. 设置广告着陆页的颜色主题

用户点击广告时，会弹出广告着陆页（或称 LandingPage 页面）。该页的动作栏颜色可供设置：



设置主题的代码：

```
// 设置'广告着陆页'动作栏的颜色主题
AppCompatActivity.setActionBarColorTheme(ActionBarColorTheme.ACTION_BAR_WHITE_THEME);
// 另外，也可设置动作栏中单个元素的颜色，颜色参数为四段制，0xFF(透明度，一般填FF)DE(红)DA(绿)DB(蓝)
//
AppCompatActivity.getActionBarColorTheme().set[Background|Title|Progress|Close]Color(0xFFDEDA DB);
```

五、横幅 Banner 广告

在Activity的onCreate函数中：

```
RelativeLayout your_original_layout = new RelativeLayout(this);
setContentView(your_original_layout);
//人群属性
AdSettings.setKey(new String[]{"baidu","中国"});
//创建广告view
String adPlaceID = "Your_adPlaceId";//重要：请填上你的代码位ID,否则无法请求到广告
adView = new AdView(this,adPlaceId);
//设置监听器
adView.setListener(new AdViewListener(){...})
//将adView添加到父控件中（注：该父控件不一定为您的根控件，只要该控件能通过addView添加广告视图即可）
RelativeLayout.LayoutParams rllp = new RelativeLayout.LayoutParams(...);
Rllp.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM)
your_original_layout.addView(adView,rllp);
```

详见 Demo 项目中的 BannerAdActivity 类

注：若您需要使用 AdViewListener 监听器，需要注意 onAdReady 函数回调时是广告资源已经缓存完毕，但是此刻广告还没有渲染出来，若需要根据广告请求状态，对广告视图可见性做切换时，可使用 onAdShow 函数。

六、插屏广告

插屏广告的调用流程与 Banner 广告不同。插屏广告需要提前预加载广告(调用 loadAd 函数)；在需要展示的时候，可以调用 showAd 函数来显示插屏广告。示例如下：

```
setContentView(R.layout.interstitialad);
String adPlaceId = "Your_adPlaceId";//重要：请填上你的代码位ID,否则无法请求到广告
interAd=new InterstitialAd(this, adPlaceId);
interAd.setListener(new InterstitialAdListener(){...});
interAd.loadAd();

Button btn=(Button)this.findViewById(R.id.btn_interstitial);
btn.setOnClickListener(new OnClickListener(){
    @Override
    public void onClick(View arg0) {
        if(interAd.isAdReady()){
            interAd.showAd(InterstitialAdActivity.this);
        }else{
            interAd.loadAd();
        }
    }
});
```

详见Demo项目中的InterstitialAdActivity类

注1：在“多次展示”插屏广告的场景中，为了让广告及时展现，推荐在InterstitialAdListener监听器的onAdDismissed方法中额外调用一次interAd.loadAd();

注2：从v4.4开始，插屏广告支持视频物料。需要在MSSP平台修改原代码位设置（开通投放视频设置）或创建新的代码位。

视频前贴片插屏图片广告

视频前贴片出现在视频播放之前，有5s倒计时，倒计时结束广告自动关闭。AdSize对应InterstitialForVideoBeforePlay。

视频暂停插屏图片广告

视频暂停贴片出现在视频播放暂停时，可以自定义大小，没有倒计时功能。AdSize对应InterstitialForVideoPausePlay。

这两种广告形式创建插屏对象使用InterstitialAd(Context context, AdSize adSize, String positionId)，加载广告使用loadAdForVideoApp(int width, int height)，展示广告使用showAdInParentForVideoApp(Activity activity, RelativeLayout parent)。广告关闭会回调接口onAdDismissed()。

注：广告请求参数width和height以及显示广告时传入的parent宽高大小要合理，不能过小。

七、开屏广告

开屏广告在您的应用启动时使用，嵌入在您的应用启动页 Activity 中，此类广告展现开始 5s 后自动关闭。

开屏是成功请求到广告后立即展现，默认可点击；可通过构造函数来设置是否接受点击类型的广告，建议设置开屏可点击，不然可能存在填充问题。构造函数如下：

```
SplashAd(Context context, ViewGroup viewParent, SplashAdListener listener, String adPlaceld, boolean canClick);
```

其中，canClick 参数表示是否接受点击类型的广告，若该参数设置为 true，广告填充率会更高。

其中，adPlaceld 参数表示广告位 id，可以为空。

开屏使用方式如下：

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.splash);
    RelativeLayout adsParent = (RelativeLayout) this
        .findViewById(R.id.adsRl);
    SplashAdListener listener=new SplashAdListener() {
        @Override
        public void onAdDismissed() {
            Log.i("RSplashActivity", "onAdDismissed");
            jumpWhenCanClick();// 跳转至您的应用主界面
        }

        @Override
        public void onAdFailed(String arg0) {
            Log.i("RSplashActivity", "onAdFailed");
            jump();
        }

        @Override
        public void onAdPresent() {
            Log.i("RSplashActivity", "onAdPresent");
        }

        @Override
        public void onAdClick() {
            Log.i("RSplashActivity", "onAdClick");
            //设置开屏可接受点击时，该回调可用
        }
    };
}

/**
```



```

* 构造函数:
* new SplashAd(Context context, ViewGroup adsParent,
*             SplashAdListener listener,String adPlaceId, boolean canClick);
*/
String adPlaceId = "Your_adPlaceId";//重要: 请填上你的代码位ID,否则无法请求到广告

new SplashAd(this, adsParent, listener, adPlaceId, true);

}

```

八、信息流广告

在百度 MSSP 后台进行 APPID 申请和代码位设置，详见 [百度 MSSP 信息流广告接入流程说明.pdf](#)。

注意：

- 一个广告只允许展现一次，多次展现、点击只会计入一次
- 广告展现前调用 `nativeResponse.isAdAvailable` 判断广告是否有效，无效广告展现不会被计入报表展现统计。

信息流广告 SDK 可在 ListView 控件中展示广告，也可以用于普通的广告区。详细使用步骤如下：

```

<!--必需: 声明 BaiduActivity, 否则可能导致计费不正常 -->
<activity android:name="com.baidu.mobad.feeds.BaiduActivity"
    android:configChanges="keyboard|keyboardHidden|orientation" />

```

1. ListView 控件中展示信息流广告

请求广告部分：

```

/**
* Step 1. 创建 BaiduNative 对象, 参数分别为:
* 上下文 context, 广告位 ID, BaiduNativeNetworkListener 监听 (监听广告请求的成功与失败)
* 注意: 请将 YOUR_AD_PLACE_ID 替换为自己的代码位 ID, 不填写无法请求到广告
*/

```

```

BaiduNative baidu = new BaiduNative(activity, YOUR_AD_PLACE_ID,
    new BaiduNativeNetworkListener() {

        @Override
        public void onNativeFail(NativeErrorCode arg0) {
            Log.w("ListViewActivity", "onNativeFail reason:"
                + arg0.name());
        }
        @Override
        public void onNativeLoad(List<NativeResponse> arg0) {
            if (arg0 != null && arg0.size() > 0) {
                nrAdList = arg0;
                showAdList();
            }
        }
    });

/**
 * Step 2. 创建requestParameters对象, 并将其传给baidu.makeRequest来请求广告
 */
RequestParameters requestParameters = new RequestParameters.Builder()
    .keywords("游戏, 职场")
    .confirmDownloading(true)// 用户点击下载类广告时, 是否弹出提示框让用户选择下载
    .build();
baidu.makeRequest(requestParameters);

```

与否

展示部分：

```

public NativeResponse getItem(int position) {
    return nrAdList.get(position);
}
public View getView(int position, View convertView, ViewGroup parent) {
    NativeResponse nrAd = getItem(position);
    if (convertView == null) {
        convertView = inflater.inflate(R.layout.native_ad_row, null);
    }
    AQuery aq = new AQuery(convertView); //用了AQuery库, 在 demo/libs 文件夹下
    aq.id(R.id.native_icon_image).image(nrAd.getIconUrl(),

```

```
        false, true);
    aq.id(R.id.native_main_image).image(nrAd.getImageUrl(),
        false, true);
    aq.id(R.id.native_text).text(nrAd.getDesc());
    aq.id(R.id.native_title).text(nrAd.getTitle());

    aq.id(R.id.native_cta).text(nrAd.isDownloadApp() ? "下载" : "查看");
    nrAd.recordImpression(convertView); //警告：调用该函数来发送展现，勿漏！
    return convertView;
}
```

详见 Demo 项目中的 `com.baidu.mobads.demo.main.feeds.ListViewActivity` 类，以及
`/res/layout/` 文件夹下面的 `native_ad_row.xml` 文件。

2. 普通广告区显示信息流广告（即 Native 原生广告）

请求广告部分：

该部分与“[ListView 中展示广告](#)”的请求广告部分类似；略

展示广告部分：

```
//use template1
AQuery aq = new AQuery(this); // 用了 AQuery 库，在 demo/libs 文件夹下
aq.id(R.id.iv_title).text(nativeResponse.getTitle());
aq.id(R.id.iv_icon).image(nativeResponse.getIconUrl());
aq.id(R.id.iv_main).image(nativeResponse.getImageUrl());
nativeResponse.recordImpression(rlTempl1); // 警告：调用该函数来发送展现，勿漏！
rlTempl1.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View view) {
        nativeResponse.handleClick(view); // 点击响应
    }
});
```

详见Demo项目中的 `com.baidu.mobads.demo.main.feeds.NativeOriginActivity` 类。

3. 信息流视频

主要类说明

Demo 项目中的 `com.baidu.mobads.demo.main.feeds.FSVideoLayout` 和 `FSVideoView` 类

- 视频播放器的实现类。如果开发者应用中无视频播放器代码，可直接将这两个类 copy 到您的工程中
- 这两个类使用到的图片资源，xml 布局文件等在 Demo 的 res 文件夹下面可以找到。
- 以下飘黄的 xml 代码示范如何在布局文件中嵌入视频播放器

```
<FrameLayout
    android:id="@+id/container"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingBottom="10dp"
    android:background="@android:color/white"
    android:layout_width="match_parent"
    android:layout_height="180dp">
    <ImageView
        android:id="@+id/big_pic"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="fitXY"/>
    <ImageView
        android:id="@+id/play_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:src="@drawable/biz_video_list_play_icon_big"/>

    <com.baidu.mobads.demo.main.feeds.FSVideoLayout
        android:id="@+id/videoview"
```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:visibility="gone" />
    </FrameLayout>

```

Demo项目中的 `com.baidu.mobads.demo.main.feeds.VideoFeedActivity` 类

- 例子Activity。示范如何使用视频播放器，进行广告播放。

请求广告部分

```

/**
 * Step 1. 创建BaiduNative对象，参数分别为：
 * 上下文context，广告位ID，BaiduNativeNetworkListener监听（监听广告请求的成功与失败）
 * 注意：请将YOUR_AD_PLACE_ID替换为自己的广告位ID
 */
BaiduNative baidu = new BaiduNative(activity, YOUR_AD_PLACE_ID,
    new BaiduNativeNetworkListener() {

        @Override
        public void onNativeFail(NativeErrorCode arg0) {
            Toast.makeText(VideoFeedActivity.this, "没有收到视频广告，请检查",
                Toast.LENGTH_LONG).show();
        }

        @Override
        public void onNativeLoad(List<NativeResponse> arg0) {
            if (arg0 != null && arg0.size() > 0) {
                nrAdList = arg0;

                mNrAd = nrAdList.get(0);

                android.util.Log.i(TAG, "mNrAd.getMaterialType() is " +
                    mNrAd.getMaterialType());
                if (mNrAd.getMaterialType() == MaterialType.VIDEO) {
                    Toast.makeText(VideoFeedActivity.this, "收到视频广告，开始播
                    放", Toast.LENGTH_LONG).show();

                    android.util.Log.i(TAG, "send message ");
                    myHandler.sendMessage(CHECK_VISIBILITY);
                } else {
                    Toast.makeText(VideoFeedActivity.this, "收到广告，但不是视频

```

```

广告，请检查", Toast.LENGTH_LONG).show());
        }
    }
}

});

/**
 * Step 2. 创建requestParameters对象，并将其传给baidu.makeRequest来请求广告
 */
RequestParameters requestParameters = new RequestParameters.Builder()
    .keywords("游戏,职场")
    .setWidth((int) (640 * density)) //需要设置请求视频的宽与高(物理像素
值)
    .setHeight((int) (360 * density))
    .confirmDownloading(true) // 用户点击下载类广告时，是否弹出提示框让用户选择下
载与否
    .build();

baidu.makeRequest(requestParameters);

```

展示广告部分

```

startConfig(mNrAd, false);
isReady = true;
try {
    showState = VisibilityChecker.getViewState(container);
    if (showState == VisibilityChecker.SHOW_STATE_SHOW) {

        mNrAd.recordImpression(container); //警告：调用该函数来发送展现，
勿漏！

        if (isAutoPlay == true) {
            mPlayBtnView.setVisibility(View.GONE);
            videoLayout.setVisibility(View.VISIBLE);
            realPlay();
        }
    } else {
        myHandler.sendMessageDelayed(CHECK_VISIBILITY, 1000);
    }
} catch (Exception e) {

```

```
// TODO Auto-generated catch block
e.printStackTrace();
}
```

点击广告部分

点击广告非视频区域（即标题，描述，icon 图标）的时候，通知 SDK 进行相应操作，跳转或者下载。

```
rl.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        mNrAd.handleClick(v, videoLayout.getCurrentPosition()); //通知SDK进行跳转
    }
});
```

或者下载

开发者如何通知 SDK

开发者需要调用如下 5 个 NativeResponse 接口的方法，以通知 SDK 当前视频的状态。否则，可能会影响计费操作。

onStart	通知 SDK，视频开始播放
onClose	通知 SDK，关闭视频
onFullScreen	通知 SDK，点击按钮至全屏播放
onComplete	通知 SDK，当次播放完成
onError	通知 SDK，视频播放出错

开发者何时调用这个 5 个方法，分为两种情况：

1. 开发者使用 Demo 的例子视频播放器。

该视频播放器提供了 FSVideoLayout.VideoPlayCallbackImpl 回调接口给开发者，开发者实现该接口，在接口实现里面分别调用 NativeResponse 的 5 个方法通知 SDK。

开发者使用下面一行代码设置回调接口。

```
videoLayout.setVideoPlayCallback(mVideoPlayCallback);
```

如下红色文字描述了回调函数的调用时机。开发者可以在回调函数中，实现自定义操作。

```
private FSVideoLayout.VideoPlayCallbackImpl mVideoPlayCallback = new
FSVideoLayout.VideoPlayCallbackImpl() {
    @Override
    public void onCloseVideo(int progress) { // 退出视频列表页面的回调函数。progress
        表示当前播放的进度，单位是毫秒
        if (mNrAd != null) {
            mNrAd.onClose(VideoFeedActivity.this, progress);
        }
    }
}
```

```

}

@Override
public void onFullScreen(int progress) { // 点击扩展至全屏播放的回调函数。progress
    // 表示当前播放的进度，单位是毫秒

    if (mNrAd != null) {
        mNrAd.onFullScreen(VideoFeedActivity.this, progress);
    }
}

@Override
public void onStart() { // 开始播放视频的回调函数
    if (mNrAd != null) {
        mNrAd.onStart(VideoFeedActivity.this);
    }
}

@Override
public void onCompletion(MediaPlayer mp) { // 播放视频完成的回调函数
    if (mNrAd != null) {
        mNrAd.onComplete(VideoFeedActivity.this);
    }
    mPlayBtnView.setVisibility(View.VISIBLE);
    videoLayout.setVisibility(View.GONE);
    if (getRequestedOrientation() == ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE) {
        videoLayout.toggleFullScreen();
    }
}

@Override
public void onError(MediaPlayer mp, int what, int extra) { // 播放视频出错的回
调函数
    if (mNrAd != null) {
        mNrAd.onError(VideoFeedActivity.this, what, extra);
    }
}
};

```

以如下代码为例，通知 SDK，点击全屏事件发生。

```

@Override
public void onFullScreen(int progress) { // 点击扩展至全屏播放的回调函数。progress
    // 表示当前播放的进度，单位是毫秒

    if (mNrAd != null) {

```



```

        mNrAd.onFullScreen(VideoFeedActivity.this, progress); // 通知SDK, 点击全
    屏事件发生
    }
}
    
```

2. 开发者不使用Demo提供的例子视频播放器，而是使用自己的播放器。

这种情况下，开发者需要在如下情况调用SDK的这5个方法。

- 视频开始播放(指的是MediaPlayer.start())的时候，调用NativeResponse的onStart(Context context)
- 视频退出的时候，调用NativeResponse的onClose(Context context, int progress)
- 全屏播放的时候，调用NativeResponse的onFullScreen(Context context, int progress)
- 视频播放完成的时候，调用NativeResponse的onComplete(Context context)
- 视频播放出错的时候，调用NativeResponse的onError(Context context, int what, int extra)

4. 信息流轮播模板

请求广告部分

```

/**
 * Step 1. 创建BaiduNative对象，参数分别为：
 * 上下文context，广告位ID，BaiduNativeNetworkListener监听（监听广告请求的成功与失败）
 * 注意：请将YOUR_AD_PLACE_ID替换为自己的广告位ID
 */
BaiduNative baidu = new BaiduNative(activity, YOUR_AD_PLACE_ID,
    new BaiduNativeNetworkListener() {

        @Override
        public void onNativeFail(NativeErrorCode arg0) {
            Toast.makeText(HTMLFeedLunBoActivity.this, "没有收到轮播模板广告，
            请检查", Toast.LENGTH_LONG).show();
        }

        @Override
        public void onNativeLoad(List<NativeResponse> arg0) {
            if (arg0 != null && arg0.size() > 0) {
    
```

```

        nrAdList = arg0;

        mNrAd = nrAdList.get(0);

        if (mNrAd.getMaterialType() == MaterialType.HTML) {
            Toast.makeText(HTMLFeedLunBoActivity.this, "收到轮播模板广
告.", Toast.LENGTH_LONG).show();

            myHandler.sendMessage(SHOW_HTML);
        } else {
            Toast.makeText(HTMLFeedLunBoActivity.this,
                "收到广告,但不是模板广告,请检查", Toast.LENGTH_LONG)
                .show();
        }
    }
}

});

/**
 * Step 2. 创建requestParameters对象, 并将其传给baidu.makeRequest来请求广告
 */
RequestParameters requestParameters = new RequestParameters.Builder()
    .keywords("游戏,职场")
    .setWidth((int) (360 * density)) // 需要设置请求模板的宽与高 (物理像素
值)
    .setHeight((int) (250 * density))
    .confirmDownloading(true) // 用户点击下载类广告时, 是否弹出提示框让用户选择下
载与否
    .build();

baidu.makeRequest(requestParameters);

```

展示广告部分

```

    if (msg.what == SHOW_HTML) {
        WebView webView = mNrAd.getWebView();
        adsParent.addView(webView); // 将SDK渲染好的WebView加入父控件
    }

```

详见 Demo 项目中的 `com.baidu.mobads.demo.main.feeds.HTMLFeedLunBoActivity` 类, 以及 `/res/layout/` 文件夹下面的 `templatefeed.xml` 文件。

5. 信息流橱窗模板

信息流橱窗模板与信息流轮播模板的 apid 不同。

请求广告部分

该部分与“[信息流轮播模板](#)”的请求广告部分类似；略

展示广告部分

该部分与“[信息流轮播模板](#)”的展示广告部分类似；略

详见 Demo 项目中的 `com.baidu.mobads.demo.main.feeds.HTMLFeedChuChuangActivity` 类,以及 `/res/layout/` 文件夹下面的 `templatefeed.xml` 文件。

九、贴片广告

1. 前贴片广告

请求广告部分

```
mAdSlotBase = (AdSlotBaseRelativeLayout) findViewById(R.id.baiduAdHolderView);
mAdSlotBase.setOnClickListener(new AdSlotBaseRelativeLayout.Listener() {

    @Override
    public void onLayoutComplete(int width, int height) {
        isAdReady = false;
        mBaiduAdContext.submitRequest(); // 发送前贴片广告请求
    }

});
...
...
mBaiduAdManager = XAdManager.getInstance(this.getApplicationContext());

mBaiduAdContext = mBaiduAdManager.newAdContext();
mBaiduAdContext.setActivity(this);
// 设置广告视频播放的 View 容器
```

```

mBaiduAdContext.setVideoDisplayBase(mAdSlotBase);
// 设置请求广告服务器的超时时间，单位为毫秒
mBaiduAdContext.setAdServerRequestingTimeout(1000);
// 设置加载广告视频资源的超时时间，单位为毫秒
mBaiduAdContext.setAdCreativeLoadingTimeout(3000);
// 设置前贴片广告的广告位 id (apid)，最大广告时长，最大广告个数。目前，前贴片广告 Demo 只支持
1 个广告视频，即后两个参数暂时无效
mBaiduAdContext.newPrerollAdSlot(BAIDU_AD_PLACEMENT_ID_OF_PREROLL, 120, 4);
    
```

展示广告部分

```

new Handler(this.getMainLooper()).post(new Runnable() {
    public void run() {
        videoPlayer.setVisibility(View.GONE);
        IAdProd slot= mBaiduAdContext.getSlotById(BAIDU_AD_PLACEMENT_ID_OF_PREROLL);
        slot.start(); // 开始播放广告视频
    }
});
    
```

SDK 如何通知开发者

SDK 需要将广告视频加载的进度，状态通知开发者，以供开发者进行下一步的处理。

开发者通过调用 `addEventListener` 方法，向 SDK 注册监听广告视频加载进度的回调函数。监听状态值如下表所示：

<code>IXAdConstants4PDK.EVENT_ERROR</code>	贴片广告加载出错
<code>IXAdConstants4PDK.EVENT_REQUEST_COMPLETE</code>	请求贴片广告结束
<code>IXAdConstants4PDK.EVENT_SLOT_STARTED</code>	该广告位播放开始
<code>IXAdConstants4PDK.EVENT_SLOT_ENDED</code>	该广告位播放结束

开发者如何通知 SDK

开发者需要及时将页面状态的变化通知 SDK，以供 SDK 做出正确的响应。否则，可能会影响计费操作。

开发者通过调用 `setActivityState` 方法将页面的状态通知 SDK，参数意义如下表所示：

<code>IXAdConstants4PDK.ActivityState.PAUSE</code>	通知 SDK，页面进入生命周期的 PAUSE 状态
--	---------------------------

IXAdConstants4PDK.ActivityState.RESUME	通知 SDK，页面进入生命周期的 RESUME 状态
IXAdConstants4PDK.ActivityState.START	通知 SDK，页面进入生命周期的 START 状态
IXAdConstants4PDK.ActivityState.STOP	通知 SDK，页面进入生命周期的 STOP 状态
IXAdConstants4PDK.ActivityState.RESTART	通知 SDK，页面进入生命周期的 RESTART 状态

详见 Demo 项目中的 `com.baidu.mobads.demo.main.basevideo.PrerollActivity` 类,以及/res/layout/文件夹下面的 : preroll.xml 文件。

十、内容联盟

备注：媒体伙伴必须在 Mssp 业务端选择接入内容联盟的应用与频道类型，以便在接入内容页中生成广告，从而获得广告收益。不进行相关操作，将无法获得内容联盟页面的广告收益。

APP 通过 SDK 接口获取展示内容联盟页面的 URL，APP 在 WebView 中加载获取到的 URL 即可展示内容联盟页面。

通过调用 SDK 接口 `getCpuInfoUrl` 获取 URL，生成的 URL 以 `onUrl` 参数形式传递给 SDK

```
CpuInfoManager.getCpuInfoUrl(this, getAppsid(), getChannel(), new UrlListener() {
    @Override
    public void onUrl(String url) {
        handleWebViewLayout(url);
    }
});
```

注意：

- 内容联盟 url 获取后只能展示一次，多次展示需要每次通过 SDK 接口重新获取

请在 Mssp(<http://mssp.baidu.com/home/developer.html>)获取频道 id

Baidu 百度 | SSP媒体服务
dftest2

首页

应用管理

APP列表

代码位管理

流量策略管理

投放限制

物料库管理

内容联盟接入

SDK/API

报告

账号与设置

首页 > 内容联盟接入

接入说明:

媒体伙伴可自主选择接入内容联盟的应用与频道类型，并下载『通用版SDK』，通过SDK接口获取展示内容联盟页面的动态URL。同时，请参照说明文档中『内容联盟』接入规则，完善动态URL参数，在webview中加载获取到的动态URL即可展示内容联盟页面。

下载SDK

接入应用: apptest 指定接入内容联盟的应用

栏目类型: 娱乐频道 效果预览

接入内容联盟

接入应用	栏目类型	APP联盟状态	创建时间
test_0510 ID: a25c32da	娱乐频道 ID: 1001	已通过	2016-05-17 16:14

十一、联系我们

请登陆网站 mssp.baidu.com，点击页面右上角帮忙中心，可查看帮助文档和进行问题咨询。



十二、注意事项

- 请务必使用在 MSSP 申请的代码位 id（以构造函数参数形式传入，adPlaceId），否则无法请求到广告
- 错误提示：控制台会输出如下信息，可以根据信息排查对应错误。
 - 如：ErrorCode: [3040002]; ErrorDesc: [广告展现失败]; Extra: [no ad];
 - 详见《Baidu_Mobads_sdk_4 错误码对照表》文档。

附件 1：申请 APP ID

1. 作用

APP ID（应用 ID）用于标识您的应用。

2. 申请流程

1. 百度 MSSP [注册](#) 并 登录 。

2. 创建应用

登录后，点击“应用管理” → “添加移动应用”，如下图所示。

新建应用

应用名称: * 请提供与应用市场中一致的应用名称

所属行业: * 请选择所属行业 正确的填写行业能够提升广告的匹配程度

系统平台: iOS Android 只能选择一种操作系统

适用终端: 全部iOS终端

接入方式: SDK API

接入方式请使用默认项（SDK）。

填写您的应用信息并提供有效的下载地址后，点击“提交”，完成应用的创建。

添加成功后，会返回上述页面，此时应用的状态为“审核中”，请耐心等待审核，审核周期为 1-2 个工作日。

3. 在应用列表中，您可以获取对应的 APP ID，请按照该用户手册中的指导进行代码的部署。

注意：只有当您的应用状态变为“审核通过”后，此时部署完代码后才会有正常的广告展现并开始计费。

4. 在效果报告页面中，您可以查看您的收益以及每个应用的详细数据。

注：本文档中的广告位 ID 即为 MSSP 上应用审核通过后的代码位 ID