NBMediation 对接说明

解压 om-sample.zip adapter 和 adapter_cn 为聚合 SDK Adapter 源码,引入到目标工程。 libs 里为三方变现 SDK, 引入到工程。 Android Studio 配置参考 demo 进行配置,文档可能有更新不及时的情况。

一、引入 SDK 步骤

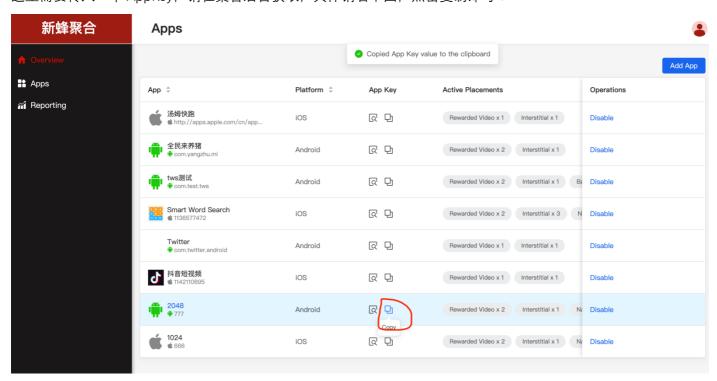
1、引入 NBMediation SDK

//NBMediation implementation(name: 'nm-android-sdk', ext: 'aar') 2、引入聚合的 SDK 和相对应的 adapter, 完全参考 demo 的 build.gradle 即可。 /tencentad cn implementation(name: 'GDTSDK.unionNormal.4.191.1061', ext: 'aar') implementation(name: 'android-query-full.0.26.7', ext: 'jar') implementation project(path: ':adapter_cn:tencentad') implementation(name: 'open_ad_sdk_2.9.5.5', ext: 'aar') implementation project(path: ':adapter_cn:tiktok')

二、聚合 SDK 初始化

请在 Application 的 onCreate 里面初始化聚合 SDK,或者在 SplashActivity 里初始化,接入开屏请参考第六章节

这里需要传入一个 AppKey, 请在聚合后台获取, 具体请看下图, 点击复制即可:



三、激励视频和插屏接入

3.1 自动广告加载

这种方式适用于使用单一广告位,并在广告位中设置瀑布流分层。此时,激励视频和插屏库存,可由 SDK 进行管理,开发者判断广告准备就绪,调用展示接口即可。瀑布流加载策略可在聚合后台配置,包括请求并发数,广告缓存池大小,请求间隔,超时时间,详细策略可咨询运营产品。

这里通过下图接口来判断,返回 true 表示激励视频或者插屏已准备就绪,可以调用 showAd 方法展示广告

设置全局激励视频的监听回调,只适用于自动管理:

设置全局插屏的监听回调,只适用于自动管理:

具体请参考 DEMO 里的 com.nbmediation.sdk.demo.MainActivity 详情

3.2 手动管理

如广告场景使用多广告位,不设置分层,可采用手动管理广告加载。开发者判断触发场景前,调用广告加载接口,加载指定广告位,并在填充回调后,调用广告展示。 插屏加载接口,注意要去掉 3.1 节中,全局插屏的监听回调。

```
//加载插屏
NmManager.getInstance().loadInterstitialAd(placementld: "235");
//设置监听回调
NmManager.getInstance().setInterstitialAdListener(placement(d: "235", new InterstitialAdListener() {
   @Override
    public void onInterstitialAdAvailabilityChanged(boolean available) {
       if (available) {
           //available为true时,可展示激励视频
    }
   @Override
    public void onInterstitialAdShowed(Scene scene) {
       NewApiUtils.printLog( msg: "onInterstitialAdShowed1 " + scene);
   @Override
    public void onInterstitialAdShowFailed(Scene scene, Error error) {
       NewApiUtils.printLog( msg: "onInterstitialAdShowFailed1 " + scene);
   @Override
    public void onInterstitialAdClosed(Scene scene) {
       NewApiUtils.printLog( msg: "onInterstitialAdClosed1 " + scene);
   @Override
    public void onInterstitialAdClicked(Scene scene) {
       NewApiUtils.printLog( msg: "onInterstitialAdClicked1 " + scene);
});
//判断激励视频是否ready
if(NmManager.getInstance().isInterstitialAdReady(placement(d: "235")){
    //展示激励视频
   NmManager.getInstance().showInterstitialAd(placementId: "235", scene: ""); //scene id可以不传
```

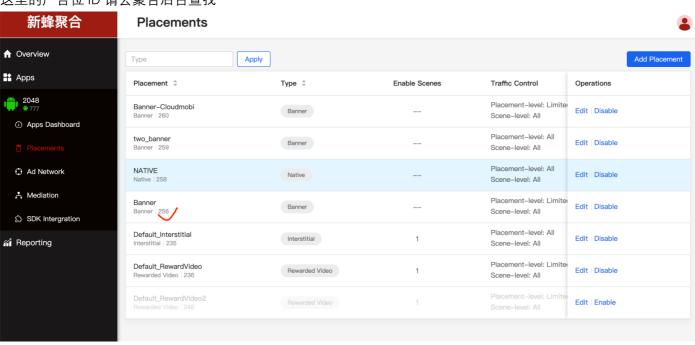
激励视频加载接口,注意要去掉3.1节中,全局激励视频的监听回调。

```
//加载激励视频
NmManager.getInstance().loadRewardedVideo(placementId: "212");
//设置监听回调
NmManager.getInstance().setRewardedVideoListener(placementId: "232", new RewardedVideoListener() {
    @Override
    public void onRewardedVideoAvailabilityChanged(boolean available) {
        if (available) {
            //available为true时,可展示激励视频
    @Override
    public void onRewardedVideoAdShowed(Scene scene) {
        NewApiUtils.printLog( msg: "onRewardedVideoAdShowed " + scene);
    @Override
    public void onRewardedVideoAdShowFailed(Scene scene, Error error) {
       NewApiUtils.printLog( msg: "onRewardedVideoAdShowFailed " + scene);
    @Override
    public void onRewardedVideoAdClicked(Scene scene) {
       NewApiUtils.printLog( msg: "onRewardedVideoAdClicked " + scene);
    @Override
    public void onRewardedVideoAdClosed(Scene scene) {
        NewApiUtils.printLog( msg: "onRewardedVideoAdClosed " + scene);
    @Override
    public void onRewardedVideoAdStarted(Scene scene) {
        NewApiUtils.printLog( msg: "onRewardedVideoAdStarted " + scene);
    @Override
    public void onRewardedVideoAdEnded(Scene scene) {
        NewApiUtils.printLog( msg: "onRewardedVideoAdEnded " + scene);
    @Override
    public void onRewardedVideoAdRewarded(Scene scene) {
        NewApiUtils.printLog( msg: "onRewardedVideoAdRewarded " + scene);
});
//判断激励视频是否ready
if(NmManager.getInstance().isRewardedVideoReady(placementld:"232")){
    //展示激励视频
    NmManager.getInstance().showRewardedVideo(placementId: "232", scene: ""); //scene id
```

四、Banner 接入

```
bannerAd = new BannerAd( activity: this, NewApiUtils.P_BANNER, new BannerAdListener() {
   .public.void.onAdReady(View.view).{
          if (null != view.getParent()) {
              ((ViewGroup) view.getParent()).removeView(view);
          adContainer.removeAllViews();
          RelativeLayout.LayoutParams.layoutParams.=.new.RelativeLayout.LayoutParams(
          adContainer.addView(view, layoutParams);
          Log.e( tag: "AdtDebug", e.getLocalizedMessage());
      bannerButton.setEnabled(true);
       bannerButton.setText("Load And Show Banner Ad");
   @Override
   .public.void.onAdFailed(String.error).{
       bannerButton.setEnabled(true);
       bannerButton.setText("Banner Load Failed, Try Again");
   .public.void.onAdClicked().{
bannerAd.setAdSize(AdSize.AD_SIZE_320X50);
bannerAd.loadAd();
```

这里的广告位 ID 请去聚合后台查找



五、native 接入

1、原生

```
nativeAd = new NativeAd( activity: this, NewApiUtils.P_NATIVE, new NativeAdListener() {
   @Override
   public void onAdFailed(String msg).{
       nativeButton.setEnabled(true);
       .nativeButton.setText("Native Load Failed, Try Again");
   @Override
   .public.void.onAdReady(AdInfo.info).{
       adView = LayoutInflater.from(MainActivity.this).inflate(R.layout.ngtive_gd_lgyout, null);
       TextView title = adView.findViewById(R.id.ad_title);
    ....title.setText(info.getTitle());
       TextView.desc.=.adView.findViewById(R.id.ad_desc);
       desc.setText(info.getDesc());
       Button btn = adView.findViewById(R.id.ad_btn);
       btn.setText(info.getCallToActionText());
       . MediaView mediaView = adView.findViewById(R.id.ad_media);
       nativeAdView = new NativeAdView( context: MainActivity.this);
       AdIconView adIconView = adView.findViewById(R.id.ad_icon_media);
       DisplayMetrics displayMetrics = MainActivity.this.getResources().getDisplayMetrics();
       mediaView.getLayoutParams().height = (int) (displayMetrics.widthPixels / (1200.0 / 627.0));
       nativeAdView.addView(adView);
```

2、Draw 信息流与内容 Draw。

draw 信息流里全部为广告;内容 draw 里会穿插内容+广告。

后台配置广告位时,添加前缀 draw 或者 cdraw,表示此广告位为 draw 和内容 draw。

SDK 依据后台配置,返回的 MediaView 即为对应的 Draw 视图,开发者将 MediavView 添加到布局上即可。根据下图参考聚合后台配置。



六、开屏接入

- 1、可新建 SplashActivity 作为 app 主 Activity
- 2、在 SplashActivity 中初始化 SDK,并在成功回调中展示开屏。