AdsGreat SDK 集成手册

SDK简介

- AdsGreat SDK支持原生、条幅、插屏、视频
- AdsGreat SDK 要求 Android API 大于15.
- AdsGreat SDK 需要获取OAID信息来提供广告展示，SDK依赖《移动智能终端补充设备标识体系统一调用 SDK》具体对接方式请参考移动安全联盟唯一官方地址：http://www.msa-alliance.cn/col.jsp?id=120 以完成对接
- 请先在AdsGreat后台添加一个app和至少一个广告位

集成AdsGreat SDK

2.1 添加SDK到工程

- jar包详情：

jar包名称	功能	必须(Y/N)
adsgreat_base_xx.jar	基础包(条幅\插屏\原生)	Y
adsgreat_imageloader_xx.jar	图片加载、缓存工具包	N
adsgreat_appwall_xx.jar	应用墙广告	N
adsgreat_video_xx.jar	视频广告	N

- 为基础包配置 build.gradle：

```
dependencies {
    implementation files('libs/adsgreat_base_xx.jar')
}
```

- 配置 AndroidManifest.xml

```
<!--Necessary Permissions-->
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />

<!-- Necessary -->
<activity android:name="com.adsgreat.base.view.InnerWebViewActivity" />



<!--If your targetSdkVersion is 28, you need update <application> as follows:-->
<application
        ...
    android:usesCleartextTraffic="true"
           ...>
           ...
</application>
```

2.2 初始化SDK

> *在应用中初始化SDK:*

请确保第一时间初始化SDK.

```
//初始化使用任意slotid即可
AdsgreatSDK.initialize(context, "Your slotID");
```

2.3 设置自定义ID

- 这里可以设置任意自定义ID，此ID会回传到我们服务器，比如激励视频的奖励回调s2s-postback，或者其它用途

```
AdsGreatVideo.setUserId("custom_id");
```

2.4 混淆配置

- 工程编译时如需混淆，请添加一下代码到progurad file:

```
#for sdk
-keep public class com.adsgreat.**{*;}
-dontwarn com.adsgreat.**

#for gaid
-keep class **.AdvertisingIdClient$** { *; }

#for js and webview interface
-keepclassmembers class * {
    @android.webkit.JavascriptInterface <methods>;
}
```

3.广告接入

我们建议您实现类AdEventListener或者EmptyAdEventListener，当getBanner、preloadInterstitialAd、getNative时重写这个方法.示例代码如下:

```
public class MyCTAdEventListener extends AdEventListener {
  /**
   * 获取广告成功
   * @param result 返回广告View，不带数据
   */
  @Override
  public void onReceiveAdSucceed(AGNative result) {
  }

  /**
   * 获取广告成功
   * @param result 纯数据对象
   */
  @Override
  public void onReceiveAdVoSucceed(AdsVO result) {
  }

  /**
   * 获取广告失败+ 数据解析失败 + 渲染失败
   */
  @Override
  public void onReceiveAdFailed(AGNative result) {
      Log.i("sdksample", "==error==" + result.getErrorsMsg());
  }

  /**
   * 广告展示成功
   */
  @Override
  public void onShowSucceed(AGNative result) {
  }


  /**
   * 成功调转到落地页
   */
```

```
    @Override
    public void onLandPageShown(AGNative result) {
    }

    /**
     * 广告被点击
     */
    @Override
    public void onAdClicked(AGNative result) {
    }

    /**
     * 广告被关闭
     */
    @Override
    public void onAdClosed(AGNative result) {
    }

}
```

3.1 原生广告

原生广告接口

> *原生广告container和layout:*

```
ViewGroup container = (ViewGroup) view.findViewById(R.id.container);
ViewGroup adLayout = (ViewGroup)View.inflate(context,R.layout.native_layout, null);
```

> *加载原生广告:*

```
/**
 * @param slotId     AdsGreat id
 * @param context    context
 * @param adListener callback listener
 */
AdsgreatSDK.getNativeAd("Your slotID", context, new MyCTAdEventListener(){
    @Override
    public void onReceiveAdSucceed(AGNative result) {
        if (result == null) {
            return;
        }
        AdvanceNative advanceNative = (AdvanceNative) result;
        showAd(advanceNative);
        ZCLog.e("onReceiveAdSucceed");
        super.onReceiveAdSucceed(result);
    }
});
```

- 展示原生广告

```

private void showAd(AdvanceNative advanceNative) {
    ImageView img = (ImageView) adLayout.findViewById(R.id.iv_img);
    ImageView icon = (ImageView) adLayout.findViewById(R.id.iv_icon);
    TextView title = (TextView)adLayout.findViewById(R.id.tv_title);
    TextView desc = (TextView)adLayout.findViewById(R.id.tv_desc);
    Button click = (Button)adLayout.findViewById(R.id.bt_click);
    ImageView adChoice = (ImageView)adLayout.findViewById(R.id.choice_icon);

    //show the image and icon yourself
    if (advanceNative.getImageFile() == null) {
        //if no cache, get image from url
        img.setImageURI(Uri.parse(advanceNative.getImageUrl()));
    } else {
        //loaded from cache
        img.setImageURI(Uri.fromFile(advanceNative.getImageFile()));
    }

    if (advanceNative.getIconFile() == null) {
        //if no cache, get image from url
        icon.setImageURI(Uri.parse(advanceNative.getIconUrl()));
    } else {
        //loaded from cache
        icon.setImageURI(Uri.fromFile(advanceNative.getIconFile()));
    }

    title.setText(advanceNative.getTitle());
    desc.setText(advanceNative.getDesc());
    click.setText(advanceNative.getButtonStr());
    adChoice.setImageURI(advanceNative.getAdChoiceIconUrl());
    //offerType (1 : download ads; 2 : content ads)
    int offerType = advanceNative.getOfferType();

    advanceNative.registeADClickArea(adLayout);
    container.addView(adLayout);
}
```

带有缓存的原生广告

> *获取缓存广告*

```
/**
 * @param slotId  slotId
 * @param context context
 * @return true for preload success,false means failure
 */
AdsgreatSDK.preloadNativeAd("Your slotID", getContext());
```

> *展示缓存的广告*

```
/**
 * @param slotId  slotId
 * @param adListener callback listener
 */
AdsgreatSDK.getNativeAdForCache("Your slotID", new MyCTAdEventListener() {
    @Override
    public void onReceiveAdSucceed(AGNative result) {
        if (result == null) return;

        AdvanceNative advanceNative = (AdvanceNative) result;
        //具体实现，请参考展示原生广告
        showAd(advanceNative);

    }
});

```

获取多个原生广告

> *调用多广告接口*

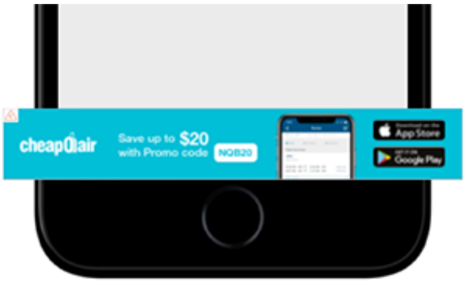
```
/**
 * @param reqAdNumber request ads num
 * @param slotId      AdsGreat id
 * @param context     context
 * @param adListener  callback listener
 */
AdsgreatSDK.getNativeAds(10, "Your slotID", getContext(), new MyCTAdEventListener(
) {

    public void onMultiNativeAdsSuccessful(List<AdvanceNative> res) {
    }

    @Override
    public void onReceiveAdFailed(AGNative result) {
    }

    @Override
    public void onAdClicked(AGNative result) {
        super.onAdClicked(result);
    }
});

```

3.2 条幅广告

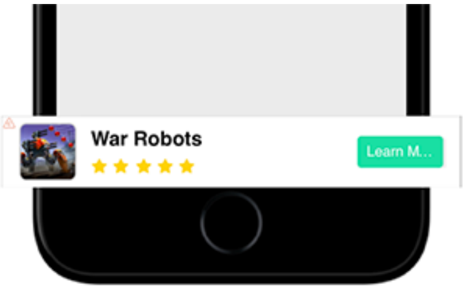
> *加载条幅广告:*

```
/**
 * Banner有3种尺寸，320*50，320*100，300*250，请参考demo BannerAdFragment.java
 * @param context    context
 * @param slotId     AdsGreat id
 * @param adSize     AdSize.AD_SIZE_320X50,
 *                   AdSize.AD_SIZE_320X100,
 *                   AdSize.AD_SIZE_300X250;
 * @param adListener   callback listener
 */
AdsgreatSDK.getBannerAd(context, "Your slotID", adSize,new MyCTAdEventListener() {

    @Override
    public void onReceiveAdSucceed(AGNative result) {
        if (result != null) {
            container.setVisibility(View.VISIBLE);
            container.removeAllViews();
            container.addView(result);   //把广告添加到容器
        }
        super.onReceiveAdSucceed(result);
    }

    @Override
    public void onReceiveAdFailed(AGNative result) {
        super.onReceiveAdFailed(result);
    }

    @Override
    public void onAdClicked(AGNative result) {
        super.onAdClicked(result);
    }

    @Override
    public void onAdClosed(AGNative result) {
        container.removeAllViews();
        container.setVisibility(View.GONE);
        super.onAdClosed(result);
    }
});
```

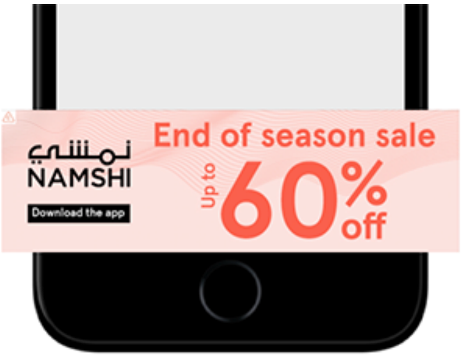
> *条幅广告集成效果如下所示：*



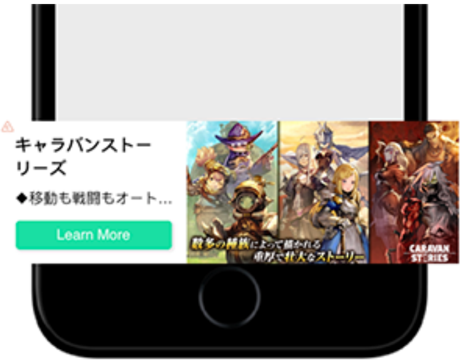
320*50



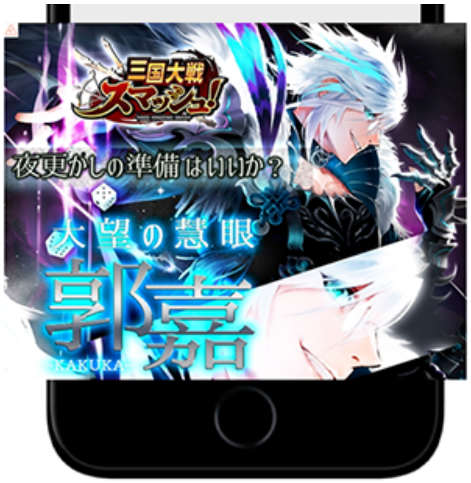
320*50



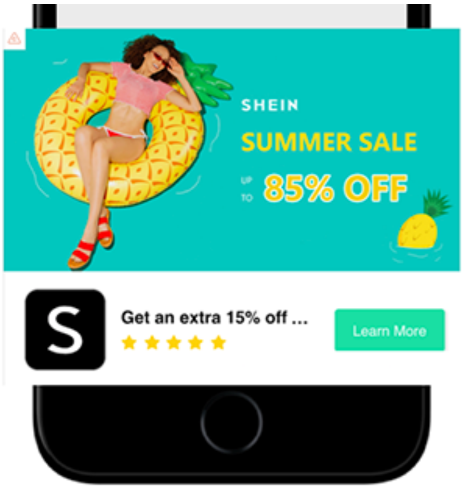
320*100



320*100



300*250



300*250

3.3 插屏广告

> *配置AndroidManifest.xml清单文件*

```
<activity android:name="com.adsgreat.base.view.InterstitialActivity" />
```

> *展示插屏广告*

```
/**
  * @param context           context
  * @param slotId            AdsGreat id
  * @param adListener        callback listener
  */
  AdsgreatSDK.preloadInterstitialAd(context, "Your slotID",new MyCTAdEventListener() {

    @Override
    public void onReceiveAdSucceed(AGNative result) {
        if (result != null && result.isLoaded()) {
            Log.w(TAG, "onReceiveAdSucceed");
            if (result.isInterstitialAvailable()) {
                result.showInterstitialAd();
           }
        }
        super.onReceiveAdSucceed(result);
    }

    @Override
    public void onLandpageShown(AGNative result) {
        super.onLandpageShown(result);
        Log.e(TAG, "onLandpageShown:");
    }


    @Override
    public void onReceiveAdFailed(AGNative error) {
        Log.w(TAG, "onReceiveAdFailed: " + error.getErrorsMsg());
        super.onReceiveAdFailed(error);
    }


    @Override
    public void onAdClosed(AGNative result) {
        super.onAdClosed(result);
        Log.e(TAG, "onAdClosed:");
    }
});
```

3.4 激励视频广告

> *配置 build.gradle 添加激励视频所需jar包：*

```
dependencies {
    implementation files('libs/adsgreat_base_xx.jar')
    implementation files('libs/adsgreat_video_xx.jar')
    implementation files('libs/adsgreat_imageloader_xx.jar')
}
```

> *配置 AndroidManifest.xml：*

```
<activity
android:name="com.adsgreat.video.view.RewardedVideoActivity"
android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|scr
eenSize|smallestScreenSize" />
```

> *预加载激励视频*
>
> *当您想向用户显示RewardedVideo广告时，最好调用此接口。这将有助于减少延迟，并确保有效、及时地发送广告。*

```
AdsGreatVideo.preloadRewardedVideo(getContext(), Config.slotIdReward,
    new VideoAdLoadListener() {

    @Override
    public void onVideoAdLoadSucceed(AGVideo videoAd) {
        video = videoAd;
        Log.w(TAG, "onVideoAdLoadSucceed: ");
    }


    @Override
    public void onVideoAdLoadFailed(AGError error) {
        Log.w(TAG, "onVideoAdLoadFailed: " + error.getMsg());
    }
});
```

> *向用户显示激励视频广告*
> *在显示视频之前，您可以通过调用来请求或查询视频状态:*

```
boolean available = AdsGreatVideo.isRewardedVideoAvailable(video);
```

> *因为加载视频创意需要一段时间。您可能需要等到视频创意完全加载。一旦获得可用的奖励视频，您就可以通过调用showRewardedVideo()方法向用户显示此视频广告，如下例所示：：*

```
AdsGreatVideo.showRewardedVideo(video, new RewardedVideoAdListener() {
    @Override
    public void videoStart() {
        Log.e(TAG, "videoStart: ");
    }

    @Override
    public void videoFinish() {
        Log.e(TAG, "videoFinish: ");
    }

    @Override
    public void videoError(Exception e) {
        Log.e(TAG, "videoError: ");
    }

    @Override
    public void videoClosed() {
        Log.e(TAG, "videoClosed: ");
    }

    @Override
    public void videoClicked() {
        Log.e(TAG, "videoClicked: ");
    }

    @Override
    public void videoRewarded(String rewardName, String rewardAmount) {
        Log.e(TAG, "videoRewarded: ");
    }
});
```

- 激励视频为单例，每个app只使用一个激励视频的slotid，加载的时候同时只有一个实例加载。当有激励视频load的时候，如果回调没有响应，此时不应再次load，需要开发者参考RewardVideoAdFragment.java的isPreload属性进行控制。

> *奖励用户*

每当用户成功完成一段视频时，SDK都会触发videoRewarded事件。RewardedVideoListener将接收此事件，因此您可以成功奖励用户。

奖励对象包含您的AdsGreat SSP中定义的SlotId的奖励名称(rewardName)和奖励金额(rewardAmount)：

```
public void videoRewarded(String rewardName, String rewardAmount) {
      //TODO - 在这里，您可以根据给定的金额奖励用户
       Log.e(TAG, "videoRewarded: ");
   }
```

4.错误码查询

Error Code	Description
ERR_000_TRACK	Track exception
ERR_001_INVALID_INPUT	Invalid parameter
ERR_002_NETWORK	Network exception
ERR_003_REAL_API	Error from Ad Server
ERR_004_INVALID_DATA	Invalid advertisement data
ERR_005_RENDER_FAIL	Advertisement render failed
ERR_006_LANDING_URL	Landing URL failed

ERR_007_TO_DEFAULT_MARKET	Default Landing URL failed
ERR_008_DL_URL	Deep-Link exception
ERR_009_DL_URL_JUMP	Deep-Link jump exception
ERR_010_APK_DOWNLOAD_URL	Application package download failed
ERR_011_APK_INSTALL	Application install failed
ERR_012_VIDEO_LOAD	Load the video exception
ERR_013_PAGE_LOAD	HTML5 page load failed
ERR_014_JAR_UPDATE_VERSION	Update jar check failed
ERR_015_GET_GAID	Cannot get google advertisement id - GAID retrieval failed
ERR_016_GET_AD_CONFIG	Cannot get the account configuration or template
ERR_017_INTERSTITIAL_SHOW_NO_AD	Tried to load the interstitial advertisement, but the advertisement is not ready/loading
ERR_018_AD_CLOSED	Ad slotId has been closed
ERR_999_OTHERS	All other errors