**SuperADS advertising SDK integration - Android**

SuperADS advertising SDK provides five different Ads formats: Banner, Interstitial, Native, Reward Video, Playable Ads. This document instruction you how to integrate the SDK for Android developers.

To take a look at Ad showing, get “asdian” APP from Google Play:

<https://play.google.com/store/apps/details?id=com.asdian.app>

code sample:

<https://github.com/156076769/superads_standalone_demo>

1. **Preparation, get a PublisherId for u and get a ApplicationId for your App**

SuperADS assign a unique id to each developer, which we call it PublisherId. Also for a APP of this developer, we assign a unique id to it, which we call it ApplicationId. Currently our operation staff will give you them, later you can login [www.superads.cn/xxx](http://www.superads.cn/xxx) to create them by yourself later.

There is also a publick key(xxx.pem) needed when your APP communicates with SuperADS server. Please contact our operator to get this key file.

1. **Dependence to SDK and Initial it**

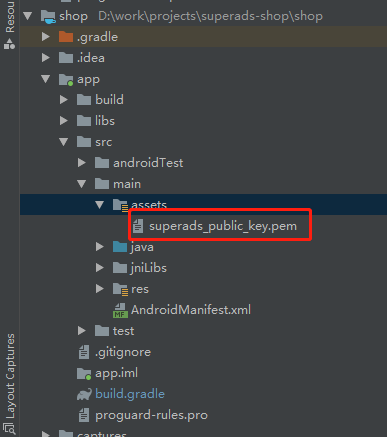
2.1 We already publish SDK library to jcenter, in your APP build.gradle, add follow to depend it

1. implementation 'com.superads.android:adsdk:0.0.7'
2. implementation 'com.github.bumptech.glide:glide:4.9.0'
3. implementation 'com.squareup.retrofit2:retrofit:2.5.0'
4. implementation 'com.squareup.retrofit2:converter-gson:2.5.0'

2.2 Proguard. Follow is the class, members, method that can’t be confused, please add them in your APP proguard rule

1. -keep **class** com.bumptech.\*\* { \*; }
2. -keep **public** **class** \* **implements** com.bumptech.glide.module.GlideModule
3. -keep **public** **enum** com.bumptech.glide.load.resource.bitmap.ImageHeaderParser$\*\* {
4. \*\*[] $VALUES;
5. **public** \*;
6. }
7. -keep **class** okhttp3.\*\* { \*; }
8. -keep **class** retrofit2.\*\* { \*; }
9. -keep **class** com.superads.android.adsdk.ads.utils.CryptoUtils { \*; }
10. -keep **class** com.superads.android.adsdk.ads.providers.SuperAds { \*; }
11. -keep **class** com.superads.android.adsdk.ads.providers.models.AdRequest { \*; }
12. -keep **class** com.superads.android.adsdk.ads.providers.models.AdRequest$Builder { \*; }
13. -keep **class** com.superads.android.adsdk.ads.rendering.view.AdView { \*; }
14. -keep **class** com.superads.android.adsdk.ads.rendering.view.InterstitialAdLoader { \*; }
15. -keep **class** com.superads.android.adsdk.ads.providers.models.NativeAdRequest { \*; }
16. -keep **class** com.superads.android.adsdk.ads.providers.models.NativeAdRequest$Builder { \*; }
17. -keep **class** com.superads.android.adsdk.ads.rendering.view.NativeAdLoader { \*; }
18. -keep **class** com.superads.android.adsdk.ads.rendering.view.VideoAdLoader { \*; }
19. -keep **class** com.superads.android.adsdk.ads.rendering.view.AdListener { \*; }
20. -keep **class** com.superads.android.adsdk.ads.providers.SuperAdsConfig { \*; }

2.3 As talk above, a pem key file is need when APP communicates with SuperADS server. This key is a RSA public key to encrypt some sensitive info in request and response. So when your get it , put it in APP assets folder, and named it “superads\_publick\_key.pem”



2.4 SDK initializtion

1. SuperAds.init(context,
2. "your publisherId", "your applicationId");

We suggest you invoke it in APP application onCreate()

1. **Request an ad and show it**

**3.1 Banner Ads**

Banner Ad is 320\*50 size picture that show in Screen top or bottom, while if screen orientation landscape, it’s a 768\*90 picture. A example is below.



How to add this ad to your app?

Step1: add a FrameLayout to contain this picture in your activity layout, and place it in top or bottom

1. <FrameLayout
2. android:id="@+id/card\_banner"
3. android:layout\_width="wrap\_content"
4. android:layout\_height="wrap\_content"
5. android:layout\_gravity="center" />

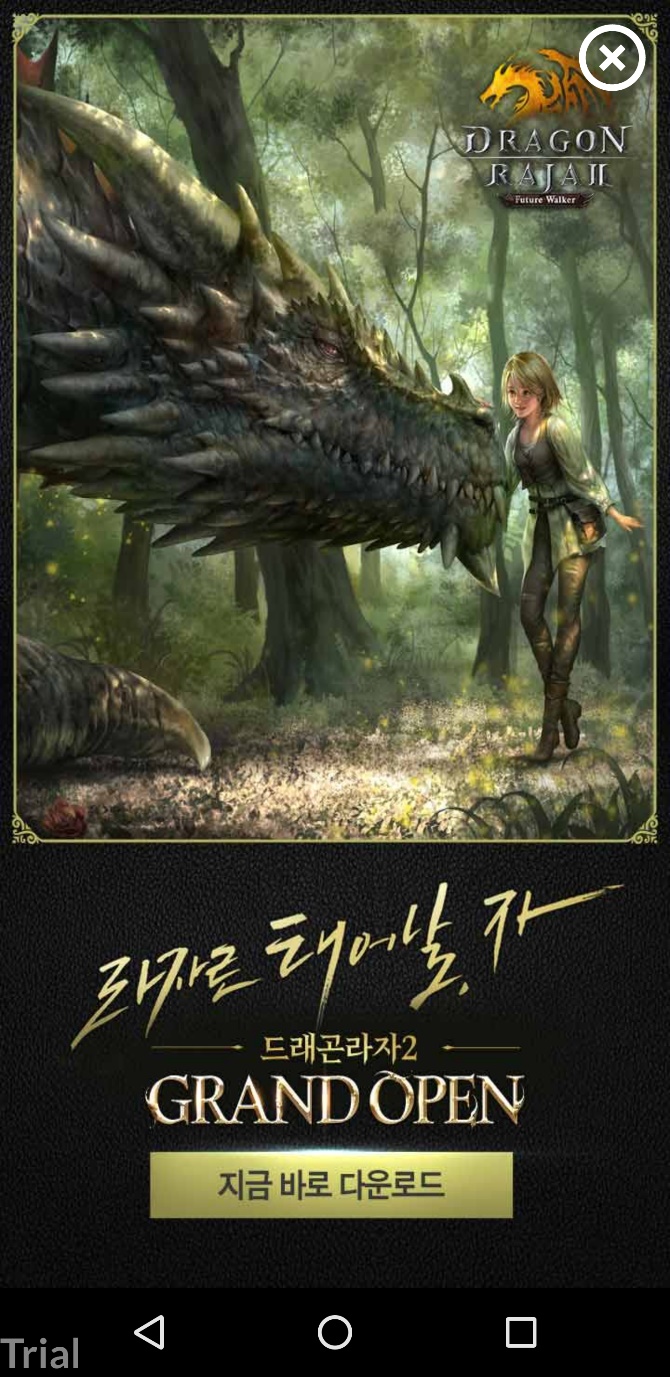
Step2: call java API to request SuperADS server, when response received, a callback will invoked and you could show view at this time.

1. **private** **void** createAndShowBanner() {
2. **final** ViewGroup bannerContainer = findViewById(R.id.card\_banner);
3. **final** AdView adView = **new** AdView(**this**);
4. AdRequest.Builder builder = **new** AdRequest.Builder(SuperAds.genRandomPlacementId());
5. adView.loadAd(builder.build(), **new** AdListener() {
6. @Override
7. **public** **void** onAdLoaded() {
8. bannerContainer.addView(adView);
9. }
11. @Override
12. **public** **void** onAdFailedToLoad(**int** errorCode) {
13. Log.e("MainActivity", "error generating ad, error code=" + errorCode);
14. }
15. });
16. }

A placement id is need when build a AdRequest, if you don’ have , you can call SuperAds.genRandomPlacementId() to generate a ramdom one.

**3.2 Interstitial Ads**

Interstitial Ad is a big picture show in full screen when opening APP or jump to another page. Below is an example.



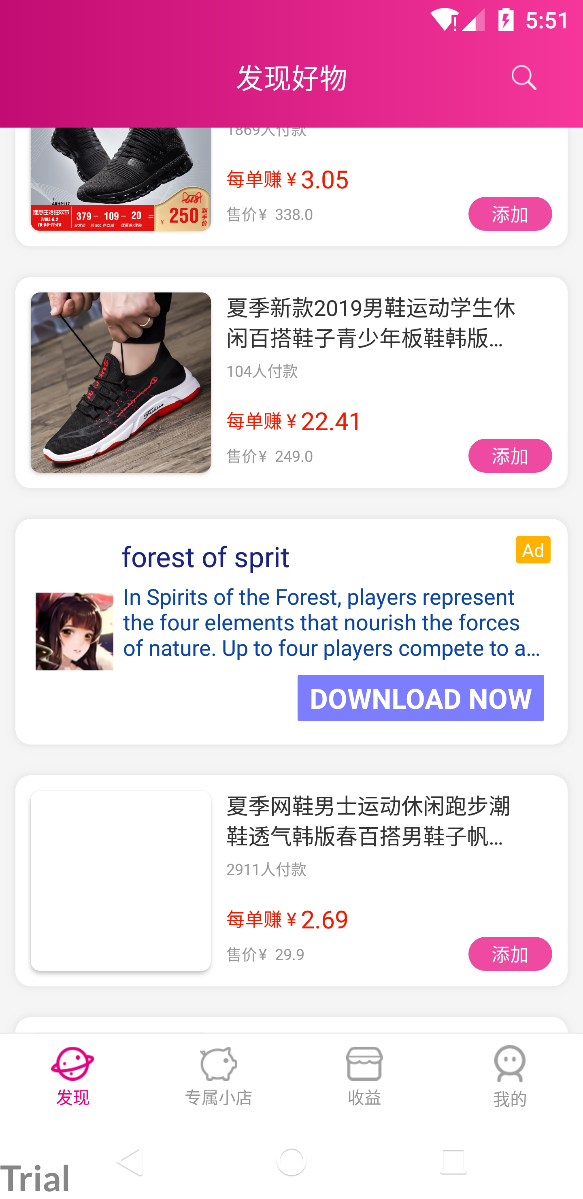
How to request and show it?

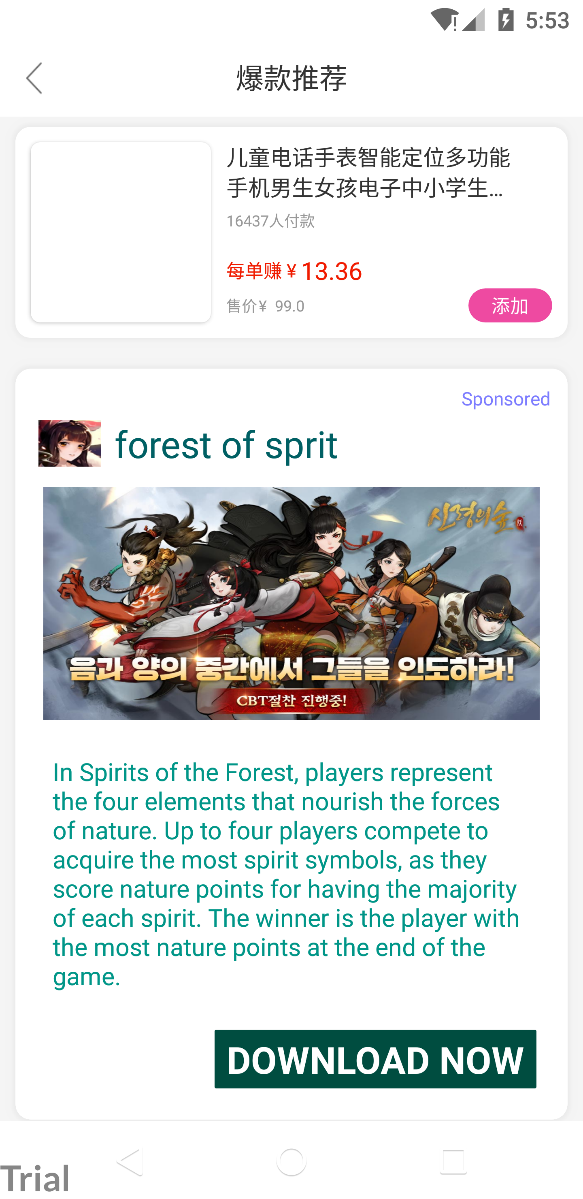
Before you jump to an activity, invoke like bellow, a full screen picture will show. After user close Ad, onAdClosed callback will invoke, here you can jump to your activity

1. AdRequest.Builder builder = **new** AdRequest.Builder("911");
2. **final** InterstitialAdLoader interstitialAd = **new** InterstitialAdLoader(getActivity());
3. interstitialAd.loadAd(builder.build(), **new** AdListener() {
4. @Override
5. **public** **void** onAdLoaded() {
6. interstitialAd.show();
7. }
9. @Override
10. **public** **void** onAdFailedToLoad(**int** errorCode) {
11. Log.e("MainActivity", "error generating ad, error code=" + errorCode);
12. }
14. @Override
15. **public** **void** onAdClosed() {
16. **super**.onAdClosed();
17. Handler handler = **new** Handler(Looper.getMainLooper());
18. handler.post(**new** Runnable() {
19. @Override
20. **public** **void** run() {
21. //jump to your activity
22. setJumpToActivity(SetInfoActivity.**class**,"关于我们",aboutUs);
23. }
24. });
25. }
26. });

**3.3 Native Ads**

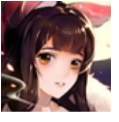
Native Ad is a RecycleView item that embed to origin list. It shows like the origin content and disturb less to user. This format is a popular currently, we provides two , Native Banner and Native Feed. Below it’s an example.





When request a Native Ad, SuperADS server responses 4(for Native Feed) or 3(for Native Banner) elements to you, they are

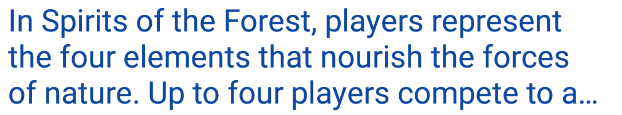
1. APP Icon：a small image that represent app icon



2. APP Title：app name



3. APP Description：description of this app, less than 120 words



4. APP Big Image：big image for APP，only Native Feed have it



Developer compose these elements together, place them appropriately, build a recycleView item, make the style same as the origin item, and then insert it to origin list.

Bellow is a example.

Step1: define a item layout that same as the origin recycleView item, contains the 4(3) elements that SuperADS server response.

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<RelativeLayout** xmlns:android="http://schemas.android.com/apk/res/android"
3. xmlns:tools="http://schemas.android.com/tools"
4. android:layout\_width="match\_parent"
5. android:layout\_height="wrap\_content"
6. android:layout\_gravity="center"
7. android:layout\_margin="0dp"
8. android:background="@color/white"
9. android:padding="0dp"**>**
11. **<TextView**
12. android:id="@+id/ad\_txt\_ad\_indicator"
13. android:layout\_width="wrap\_content"
14. android:layout\_height="wrap\_content"
15. android:layout\_alignParentTop="true"
16. android:layout\_alignParentEnd="true"
17. android:layout\_alignParentRight="true"
18. android:layout\_margin="1dp"
19. android:gravity="center"
20. android:text="Sponsored"
21. android:textColor="@color/blue"
22. android:textSize="12sp"
23. tools:ignore="HardcodedText" **/>**
25. **<ImageView**
26. android:id="@+id/ad\_img\_icon"
27. android:layout\_width="40dp"
28. android:layout\_height="40dp"
29. android:layout\_alignTop="@+id/ad\_txt\_title"
30. android:layout\_alignBottom="@+id/ad\_txt\_title"
31. android:layout\_margin="5dp"
32. android:scaleType="fitXY" **/>**
34. **<TextView**
35. android:id="@+id/ad\_txt\_title"
36. android:layout\_width="wrap\_content"
37. android:layout\_height="wrap\_content"
38. android:layout\_below="@+id/ad\_txt\_ad\_indicator"
39. android:layout\_toEndOf="@+id/ad\_img\_icon"
40. android:layout\_toRightOf="@+id/ad\_img\_icon"
41. android:padding="4dp"
42. android:textColor="#006064"
43. android:textSize="24sp"
44. tools:text="Awesome application that you have to try is now available for free download" **/>**
46. **<ImageView**
47. android:id="@+id/ad\_img"
48. android:layout\_width="match\_parent"
49. android:layout\_height="150dp"
50. android:layout\_below="@+id/ad\_txt\_title"
51. android:layout\_centerHorizontal="true"
52. android:layout\_margin="8dp"
53. android:scaleType="fitXY" **/>**
55. **<TextView**
56. android:id="@+id/ad\_txt\_description"
57. android:layout\_width="wrap\_content"
58. android:layout\_height="wrap\_content"
59. android:layout\_below="@+id/ad\_img"
60. android:layout\_margin="12dp"
61. android:ellipsize="end"
62. android:padding="2dp"
63. android:textColor="#009688"
64. android:textSize="16sp"
65. tools:text="Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat." **/>**
67. **<TextView**
68. android:id="@+id/ad\_txt\_cta"
69. android:layout\_width="wrap\_content"
70. android:layout\_height="wrap\_content"
71. android:layout\_below="@id/ad\_txt\_description"
72. android:layout\_alignParentEnd="true"
73. android:layout\_alignParentRight="true"
74. android:layout\_margin="8dp"
75. android:background="@drawable/cta\_square\_green"
76. android:gravity="center"
77. android:paddingStart="10dp"
78. android:paddingLeft="10dp"
79. android:paddingTop="5dp"
80. android:paddingEnd="10dp"
81. android:paddingRight="10dp"
82. android:paddingBottom="5dp"
83. android:textAllCaps="true"
84. android:textColor="@color/white"
85. android:textSize="24sp"
86. android:textStyle="bold"
87. tools:text="Download now" **/>**
89. **</RelativeLayout>**

Step2: build a ViewHolder for Native Ad that will insert into recycleView adapter. When building, asynchronous request Ad, add ViewHodler view to recycleView when a response callback invoked.

1. @Override
2. **public** RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {
3. **if** (viewType !=  NORMAL) {
4. **final** View v = LayoutInflater.from(parent.getContext())
5. .inflate(R.layout.native\_banner\_ad, parent, **false**);
6. **final** View adView;
7. NativeAdRequest.Builder builder;
8. **if** (nativeAdType == NATIVE\_AD\_BANNER) {
9. adView = LayoutInflater.from(parent.getContext()).inflate(R.layout.native\_ad\_example\_banner\_ad, **null**);
10. builder = **new** NativeAdRequest.Builder(adView, SuperAds.genRandomPlacementId())
11. .titleTextViewId(R.id.ad\_txt\_title)
12. .descriptionsTextViewId(R.id.ad\_txt\_description)
13. .callToActionTextViewId(R.id.ad\_txt\_cta)
14. .iconImageViewId(R.id.ad\_img\_icon);
15. } **else** {
16. adView = LayoutInflater.from(parent.getContext()).inflate(R.layout.native\_ad\_example\_feed\_ad, **null**);
17. builder = **new** NativeAdRequest.Builder(adView, SuperAds.genRandomPlacementId())
18. .titleTextViewId(R.id.ad\_txt\_title)
19. .descriptionsTextViewId(R.id.ad\_txt\_description)
20. .callToActionTextViewId(R.id.ad\_txt\_cta)
21. .bigImageViewId(R.id.ad\_img)
22. .iconImageViewId(R.id.ad\_img\_icon);
23. }
25. **final** NativeAdLoader nativeAd = **new** NativeAdLoader();
26. nativeAd.loadAd(builder.build(), **new** AdListener() {
27. @Override
28. **public** **void** onAdLoaded() {
29. Log.i("NativeBannerAdapter", "native ad loaded");
30. }
32. @Override
33. **public** **void** onAdFailedToLoad(**int** errorCode) {
34. Log.e("NativeBannerAdapter", "error generating ad, error code=" + errorCode);
35. v.setVisibility(View.GONE);
36. RecyclerView.LayoutParams param = (RecyclerView.LayoutParams)v.getLayoutParams();
37. param.height = 0;
38. param.width = 0;
39. }
40. });
41. **return** **new** NativeAdViewHolder(v, adView);
42. } **else** {
43. MyViewHolder myViewHolder;
44. View view = LayoutInflater.from(context).inflate(R.layout.adapter\_all\_shop, parent, **false**);
45. myViewHolder = **new** MyViewHolder(view);
46. **return** myViewHolder;
47. }
48. }

**3.4 Video Ads**

Video Ads is a short video that plays to user when jumping Activities, it will show in full screen.

1. AdRequest.Builder builder2 = **new** AdRequest.Builder(SuperAds.genRandomPlacementId());
2. **final** VideoAdLoader videoAdLoader = **new** VideoAdLoader(getActivity());
3. videoAdLoader.loadAd(builder2.build(), **new** AdListener() {
4. @Override
5. **public** **void** onAdLoaded() {
6. Log.d("####", "### VideoAdLoader onAdLoaded");
7. videoAdLoader.show();
8. }
10. @Override
11. **public** **void** onAdFailedToLoad(**int** errorCode) {
12. Log.e("MainActivity", "error generating ad, error code=" + errorCode);
13. Handler handler = **new** Handler(Looper.getMainLooper());
14. handler.post(**new** Runnable() {
15. @Override
16. **public** **void** run() {
17. setJumpToActivity(SetInfoActivity.**class**,"about us","");
18. }
19. });
20. }
22. @Override
23. **public** **void** onAdClosed() {
24. **super**.onAdClosed();
25. Handler handler = **new** Handler(Looper.getMainLooper());
26. handler.post(**new** Runnable() {
27. @Override
28. **public** **void** run() {
29. setJumpToActivity(SetInfoActivity.**class**,"about us","");
30. }
31. });
32. }
33. });