**# Android SDK – SuperADS**

\*\*Latest Version:\*\* 1.2.6

\*\*Release Date:\*\* 17.10.19

This SuperADS SDK allows you to add several Advertisements to your project.

From Banner Ad, Interstitial Ad, Video Ad, rewareded Ad and Native Ad.

This SDK can work on Android phones and tablets

**## Requirements**

Android Studio 3.4+

Android device or emulator with Android SDK v4.1+

### Table of Contents

1. [ Setup - Add SuperADS SDK to your project ](#1)

2. [ Integrate the SDK with your code ](#2)

3. [ Types of Advertisements (examples) ](#3)

4. [ Demo app ](#4)

<a name="1"></a>

**## Setup**

Add the following to your dependencies section in module's build.grade file:

Project build.gradle

allprojects {

repositories {

google()

jcenter()

maven {

url 'https://dl.bintray.com/superads/maven'

}

}

}

App build.gradle

dependencies {

implementation 'cn.superads:sdk:1.2.6'

}

<a name="2"></a>

**## Integrate the SDK with your code**

- \*\*Initialize\*\* the SDK in your code.

This code should be called only once in app life cycle. Best to call it from MainAcitivity.onCreate:

(\*\*your publisher id\*\* is chosen by the user).

SuperAds.initialize(this, <Your\_publisher\_id>);

<a name="3"></a>

**## Types of Advertisements (examples)**

**- ### Banner Ad**

##### Banner ads usually appear at the top or bottom of your app’s screen. Adding one to your app takes just a few lines of code.

Create banner ad instance:

AdSize adSize = new AdSize("320","50"); // Size of the banner

SAAdCard adCard = SuperAds.getInstance().createBannerAdCard(adSize, this);

Optional parameter is the ad is playable:

```java

adCard.setContentType(AdContentType.PLAYABLE);

```

Set placement ID (from dashboard):

```java

adCard. setAdUnitId("<ad\_unit\_id>")

```

Ad container view:

```java

bannerContainer:

<FrameLayout

android:id="@+id/banner\_container"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginBottom="30dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

/>

```

Create callback for events:

```java

adCard.setAdListener(new SAAdListener() {

@Override

public void onAdLoaded(SAAdCard card) {

bannerContainer.addView(adCard.getAdView()); //Adding the view to container to show on the screen

}

@Override

public void onAdFailedToLoad(int errorCode, SAAdCard card) {

Logger.e("Error generating ad, error code=" + errorCode);

}

});

Load Ad:

adCard.load()

**- ### Interstitial Ad**

##### Interstitial ads provide full-screen experiences, commonly incorporating rich media to offer a higher level of interactivity compared to banner ads. Interstitials are typically shown during natural transitions in your app; for example, after completing a game level, or while waiting for a new view to load.

Create interstitial ad instance:

```java

AdSize adSize = new AdSize("768","1024"); // Size of the banner

SAAdCard adCard = SuperAds.getInstance().createInterstitialAdCard(adSize, this);

```

Optional parameter is the ad is playable:

```java

adCard.setContentType(AdContentType.PLAYABLE);

```

Set placement ID (from dashboard):

```java

adCard. setAdUnitId("<ad\_unit\_id>")

```

Create callback for events:

```java

adCard.setAdListener(new SAAdListener() {

@Override

public void onAdLoaded(SAAdCard card) {

adCard.show(); // Show the ad

}

@Override

public void onAdFailedToLoad(int errorCode, SAAdCard card) {

Logger.e("Error generating ad, error code=" + errorCode);

}

});

```

Load Ad:

adCard.load()

**- ### Video Ad**

##### A full screen video advertisement that can be shown during app use.

Create video ad instance:

```java

AdSize adSize = new AdSize("1280","720"); // Size of the video

SAAdCard adCard = SuperAds.getInstance().createVideoAdCard(adSize, this);

```

Set placement ID (from dashboard):

```java

adCard. setAdUnitId("<ad\_unit\_id>")

```

Create callback for events:

```java

adCard.setAdListener(new SAAdListener() {

@Override

public void onAdLoaded(SAAdCard card) {

adCard.show(); // Start playing the video ad

}

@Override

public void onAdFailedToLoad(int errorCode, SAAdCard card) {

Logger.e("Error generating ad, error code=" + errorCode);

}

@Override

public void onVideoCompleted(SAAdCard card) {

Logger.i("Video completed.");

}

});

```

Load Ad:

adCard.load()

**- ### Rewarded Video Ad**

##### Rewarded video ads are a great way to keep users engaged in your app while earning ad revenue. The reward generally comes in the form of in-game currency (gold, coins, power-ups, etc.) and is distributed to the user after a successful video completion.

Create video ad instance:

```java

AdSize adSize = new AdSize("1280","720"); // Size of the video

SAAdCard adCard = SuperAds.getInstance().createRewardedVideoAdCard(adSize, this);

```

Set placement ID (from dashboard):

```java

adCard. setAdUnitId("<ad\_unit\_id>")

```

Create callback for events:

```java

adCard.setAdListener(new SAAdListener() {

@Override

public void onAdLoaded(SAAdCard card) {

adCard.show(); // Start playing the video ad

}

@Override

public void onAdFailedToLoad(int errorCode, SAAdCard card) {

Logger.e("Error generating ad, error code=" + errorCode);

}

@Override

public void onVideoCompleted(SAAdCard card) {

Logger.i("Video completed.");

}

@Override

public void onVideoRewarded(SAAdCard card) {

Logger.i("Video rewarded!");

}

});

```

Load Ad:

adCard.load()

**- ### Native Ad**

##### Native ads make it easy for you to monetize your app in a way that’s consistent with its existing design. The SuperADS SDK gives you access to an ad’s individual assets so you can design the ad layout to be consistent with the look and feel of your app.

Create NativeAd instance:

```java

SAAdCard adCard = SuperAds.getInstance().createNativeAdCard(this);

```

```java

adCard. setAdUnitId("<ad\_unit\_id>")

```

Set UI fields to fill in after ad is loaded (optional):

```java

adCard.titleTextViewId(R.id.ad\_txt\_title)

adCard.privacyInformationIconImageId(R.id.privacy\_icon\_2)

adCard.descriptionsTextViewId(R.id.ad\_txt\_description)

adCard.callToActionTextViewId(R.id.ad\_txt\_cta)

adCard.iconImageViewId(R.id.ad\_img\_icon)

adCard.bigImageViewId(R.id.ad\_img)

```

Create callback for events:

```java

adCard.setAdListener(new SAAdListener() {

@Override

public void onAdLoaded(SAAdCard card) {

adCard.show(); // Show just loaded ad

}

@Override

public void onAdFailedToLoad(int errorCode, SAAdCard card) {

Logger.e("Error generating ad, error code=" + errorCode);

}

});

```

Load Ad:

adCard.load()

**## List of events:**

##### Those events can be found on SaAdListener interface

```java

void onAdLoaded(SAAdCard card) // Ad was loaded successfully and ready to be shown

void onAdFailedToLoad(int errorCode, SAAdCard card)

void onAdClosed(SAAdCard card)

void onAdLeftApplication(SAAdCard card)

void onAdOpened(SAAdCard card)

void onAdClicked(SAAdCard card)

void onAdImpression(SAAdCard card)

void onVideoCompleted(SAAdCard card)

void onVideoRewarded(SAAdCard card)

```

<a name="4"></a>

## Demo App

##### A demo app with a simple usage of the SDK can be found in the following

Github repository: [Link](https://github.com/156076769/ads-sdk-publish/tree/master/superads-android-sdk-sample "Link")

The demo app contains several types of advertisements:

\*\*Banner\*\* Ad, \*\*Interstitial\*\* Ad, \*\*Video\*\* Ad, \*\*Rewarded Video\*\* Ad and a \*\*Native\*\* Ad.