

Mobile Android SDK Usage Guide (v1.1.0)

- 1. Prerequisites (선행 요건)
- 2. Configure your app (앱 설정)
- 3. Config Affiliate ID & Sub ID in AndroidManifest.xml
- 4. Create AdViewModel(광고 로딩)
 - 4.1. Parameters description
 - 4.1.1. Supported Creative Sizes
- 5. Load Ads Data
- 6. Configure and display the Ads
 - 6.1. Static Banner
 - 6.1.1. Add AdsBannerView to the layout (레이아웃에 AdsBannerView 추가)
 - 6.1.2. Bind AdViewModel to AdsBannerView
 - 6.2. Auto Scroll Banner
 - 6.2.1. Add AutoScrollBannerView to the layout (레이아웃에 AutoScrollBannerView 추가)
 - 6.2.2. Create AdViewModel with AdModel.SCROLL parameter
 - 6.2.3. Bind ViewModel to AutoScrollBannerView
 - 6.3. Interstitial Ads
 - 6.3.1. Create an Interstitial object when Ad loaded.
 - 6.3.2. Show the interstitial ad.
- 7. Error codes (에러 코드)
- 8. Permission update required by Android 13 update (안드로이드 13 업데이트에 따른 퍼미션 업데이트 안내)

1. Prerequisites (선행 요건)

To use the Coupang Ads SDK, an account need to be registered on Coupang Partners (<https://partners.coupang.com/#help/partners-guide>), and the following parameters need to be available:

쿠팡 광고 SDK 사용을 위해서는 쿠팡 파트너스 (<https://partners.coupang.com/#help/partners-guide>) 에 계정을 생성, 등록해야 하며, 아래 파라미터가 필요하다.

- Affiliate ID (required)/ 어필리에이트 ID(필수)
 - Coupang Partners subscription ID (AF + 7 digits)
 - 쿠팡 파트너스 가입 아이디 (AF+7자리 숫자로 구성)
- Sub ID (required)/ 서브 ID(필수)
 - 8~20 characters with numbers or english case letters. When registering Sub ID, it must match the information registered in Channel ID in Partners. Up to 10 available
 - 8~20자 숫자 및 영어대소문자. Sub ID 등록 시 파트너스 내 "채널 아이디"에 등록된 정보와 반드시 매칭 되어야 함. 최대 10개까지 등록 가능
- Widget ID (required)/ 위젯 ID(필수)
 - Please refer to Coupang Partner's documentation on how to create a Dynamic Banner and obtain a Widget ID. A widget id is required to load data into an ad view.
 - 쿠팡 파트너스의 다이내믹 배너 생성 가이드를 참고하여 위젯 ID를 생성한다. 광고 노출을 위한 데이터 로딩을 위해서는 위젯 ID가 필요하다.
 - Widget ID banner setting option: "Banner Type > Customer Interest Base Recommendation" & "Banner Data > Recommended Product Feed"
 - 위젯ID 배너 설정 옵션 : "배너타입 > 고객 관심 기반 추천" & "배너 데이터 > 추천 상품 피드" 로 등록

2. Configure your app (앱 설정)

Add the following options in your project-level "build.gradle" file

아래 옵션을 프로젝트별 'build.gradle' 파일에 추가한다.

```
repositories {  
    maven { url "https://raw.githubusercontent.com/coupang-ads-sdk/android/main" }  
}
```

Add the dependencies for the Mobile Ads SDK to your module's app-level Gradle file, which is usually "app/build.gradle"

모바일 광고 SDK의 경우 모듈 앱 단위 Gradle 파일(통상 'app/build.gradle')에 디펜던시를 추가한다.

```
dependencies {  
    implementation 'com.coupang:ads:1.1.0'  
}
```

If you need to release. Then please configure this code in "proguard-rules.pro"
릴리즈 시 해당 코드를 'proguard-rules.pro'에 설정한다.

```
-keep interface com.coupang.ads.dto.DTO
-keep class * implements com.coupang.ads.dto.DTO { *; }
```

3. Config Affiliate ID & Sub ID in AndroidManifest.xml

Before using the SDK, config the Affiliate ID & Sub ID in your AndroidManifest.xml

SDK 사용 전, 어필리에이트 ID 신청이 되어 있어야 한다. 또한, AdsBannerView를 사용하기 전에 SDK를 초기화하고, 어필리에이트 ID를 제출해야 한다.

AndroidManifest.xml

```
<application>
  <meta-data android:name="coupang_ads_affiliate_id" android:value="{your Affiliate ID}"/>
  <meta-data android:name="coupang_ads_sub_id" android:value="{your Sub ID}"/>
</application>
```

4. Create AdViewModel(광고 로딩)

Create an AdViewModel with widgetId, AdCreativeSize and other optional parameters.

Kotlin example :

Creative AdViewModel in kotlin

```
class KtActivity: AppCompatActivity() {
    ...
    private val bannerAdViewModel: AdViewModel by createAdViewModel(
        "514017", // your widgetId
        AdCreativeSize._320x100,
        affiliatePage = "MainPage",
        affiliatePlacement = "Bottom Banner"
    )
    ...
}
```

Java example:

```

import com.coupang.ads.tools.ViewModelExtensionsKt;

public class JavaActivity extends AppCompatActivity {
    private bannerAdsViewModel AdsViewModel;

    @Override
    protected void onCreate(@Nullable @org.jetbrains.annotations.Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // NOTE: Make sure to call this only after the super.onCreate(savedInstanceState) above.
        bannerAdsViewModel = ViewModelExtensionsKt.createAdsViewModelJava(
            this,
            AdsViewModel.class,
            "514017", // your widgetId.
            AdsCreativeSize._320x100,
            affiliatePage = "MainPage",
            affiliatePlacement = "Bottom Banner"
        );
    }
}

```

Note: The `AdsViewModel` can also be created inside `Fragment` class.

4.1. Parameters description

The creation of `AdsViewModel` supports the following parameters:

param name	Description	Required?
widgetId	WidgetId for the ad. See the Prerequisites section above for details.	Yes
adsCreativeSize	Size of the ad. See the 'Supported Creative Sizes' section below for details	Yes
affiliatePage	The page name in the app 광고 페이지명	No
affiliatePlacement	the position of the ad in the page 페이지 상 광고의 노출 위치	No
adsMode	Parameter to control the ads display mode, default is <i>AdsMode.AUTO</i> . Note: For auto scroll ads, set this value to <i>AdsMode.SCROLL</i>	No

4.1.1. Supported Creative Sizes

AdsCreativeSize enum (CreativeSize 상수)	Size in DP unit (WxH)
AdsCreativeSize._320x50	320x50
AdsCreativeSize._320x100	320x100
AdsCreativeSize._300x250	300x250

AdsCreativeSize.SMART	<p>Screen width (화면 너비) x 32 50 90</p> <p>Screen width (화면 너비) <=400dp height (높이)=32</p> <p>400dp<Screen width (화면 너비) <=720dp height (높이)=50</p> <p>Screen width (화면 너비) >720dp height (높이)=90</p>
AdsCreativeSize.INTERSTITIAL	Screen Width x Screen Height (i.e. This is a full screen ad)

5. Load Ads Data

To load Ad from Coupang, call the `loadAdData()` of the `adsViewModel` instance created above. You can also *observe* the data loading result or possible errors and perform your action, see example below.

Kotlin example:

LoadAdData with AdsViewModel in kotlin

```
// Observe ads loading results.
bannerAdsViewModel.observe(this) {
    if (it.isSuccess) {
        // Code to be executed when an ad finishes loading.
        // e.g. Log.i("observe", "loadAdData success")
    } else {
        // Code to be executed when ad failed to load.
        // e.g. Log.w("observe", "loadAdData filed", it.exceptionOrNull())
    }
}

// Load ads data
bannerAdsViewModel.loadAdData()
```

Java Example:

LoadAdData with AdsViewModel in Java

```
import com.coupang.ads.java.Result;

// Observe ad loading results.
bannerAdsViewModel.observe(this, result -> {
    Result<Product> r = new Result<>(result);
    if (r.isSuccess()) {
        // Code to be executed when an ad finishes loading.
        // e.g. Log.i("observe", "loadAdData success")
    } else {
        // Code to be executed when ad failed to load.
        // e.g. Log.w("observe", "loadAdData filed", it.exceptionOrNull())
    }
});

// Load ads data.
bannerAdsViewModel.loadAdData();
```

6. Configure and display the Ads

Three styles of ads are supported: Static Banner, Auto Scroll Banner, and Interstitial Ads. Their configurations are different, see details below

6.1. Static Banner

6.1.1. Add AdsBannerView to the layout (레이아웃에 AdsBannerView 추가)

layout

```
# main_activity.xml
...
<com.coupang.ads.view.banner.AdsBannerView
    android:id="@+id/ad_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
...
```

6.1.2. Bind AdViewModel to AdsBannerView

Kotlin example:

```
findViewById(R.id.ad_view).bindViewModel(this, bannerAdViewModel)
```

6.2. Auto Scroll Banner

6.2.1. Add AutoScrollBannerView to the layout (레이아웃에 AutoScrollBannerView추가)

layout

```
# main_activity.xml
...
<com.coupang.ads.view.banner.auto.AutoScrollBannerView
    android:id="@+id/auto_scroll_banner"
    android:layout_width="match_parent"
    android:layout_height="150dp" />
...
```

6.2.2. Create AdViewModel with AdsModel.SCROLL parameter

Kotlin example:

Creative AdViewModel in kotlin

```
class KtActivity: AppCompatActivity() {
    ...
    private val scrollBannerAdViewModel: AdViewModel by createAdViewModel(
        "514017", // your widgetId
        AdsCreativeSize._320x100,
        affiliatePage = "MainPage",
        affiliatePlacement = "Bottom Banner",
        adsMode = AdsMode.SCROLL
    )
    ...
}
```

6.2.3. Bind ViewModel to AutoScrollBannerView

Kotlin example:

```
findViewById(R.id.auto_scroll_banner).bindViewModel(this, scrollBannerAdViewModel)
```

6.3. Interstitial Ads

Interstitial is a type of full screen ads, which can only be dismissed when user click the *close* button at the top right corner. Follow these steps to display the interstitial ads:

6.3.1. Create an Interstitial object when Ad loaded.

Kotlin Example:

```
class KtActivity: AppCompatActivity() {

    private val interstitialViewModel: AdsViewModel by adsViewModels(
        "514017", // your widget id.
        CreativeSize.INTERSTITIAL
    )

    private val interstitialAd by lazy {
        AdsInterstitial().also {
            it.listener = object : AdsInterstitialListener {
                override fun onAdFailedToShow(e: AdsException) {
                    // Code when ad failed to show.
                    Log.e("interstitial", "onShowFailed", e)
                }

                override fun onAdShowed() {
                    // Code when Ad is showed successfully.
                    Log.d("interstitial", "onShow")
                }

                override fun onAdDismissed() {
                    // Code when ad is dismissed
                    Log.d("interstitial", "onDismiss")
                }
            }
        }
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        ...

        interstitialAd.bindViewModel(interstitialViewModel)
        interstitialViewModel.loadAdData()
    }
}
```

6.3.2. Show the interstitial ad.

Interstitial ads should be displayed during natural pauses in the flow of an app. Between levels of a game is a good example, or after the user completes a task. To show an interstitial, use the `showAds()` method.

Kotlin example:

```
class KtActivity: AppCompatActivity() {
    ...
    private fun showInterstitial() {
        if (interstitialAd.isAvailable()) {
            interstitialAd?.showAds(this)
        } else {
            Log.d("interstitial", "The interstitial ad wasn't available yet.")
        }
    }
    ...
}
```

7. Error codes (에러 코드)

Error codes	Solution
Data is empty	This error means Coupang ads server didn't return any ad for the request. This usually happens in two cases: 1) The device couldn't be mapped to any Coupang user. In this case, please install Coupang App and login in the same tested device. For the first time you installed Coupang app and logged in, please wait for one day to retry. 2) The user has opted out for Coupang personalized ads. In this case, please login Coupang App with another non opted-out account and retry after one day.
Ad id is empty or Failed to get Android ADID	This error means the android AdId for advertising (a.k.a AAID) can't be retrieved. This might happen in two cases: 1) The user has explicitly opted-out personalized ads in the device. If you still want to test with that device, please opt in and retry. 2) If the user just installed the app, there might be a small possibility they get this one time error when using the app for this first time. This will happen only once, please ignore this error (i.e. don't show the ad in this case).

8. Permission update required by Android 13 update (안드로이드 13 업데이트에 따른 퍼미션 업데이트 안내)

Due to Google's policy change, Android 13 requires additional permissions to collect ADIDs, so please be sure to add the following permissions to AndroidManifest.xml if you have plan to upgrade your app to target Android 13 (API 33) device or above. See this link for more details: [https://developers.google.com/android/reference/com/google/android/gms/ads/identifier/AdvertisingIdClient.Info#getId\(\)](https://developers.google.com/android/reference/com/google/android/gms/ads/identifier/AdvertisingIdClient.Info#getId())

```
<uses-permission android:name="com.google.android.gms.permission.AD_ID" />
```