

Mobile Android SDK Usage Guide (v1.0.2)

1.1. 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

1.2. 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:1.0.2'  
}
```

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 { *; }
```

1.3. Init Coupang SDK (쿠팡 SDK 초기화)

Before using the SDK, you must have applied for the affiliate id. at the same time, before using `AdsBannerView`, you should initialize the SDK and pass in the applied affiliate id.

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

Init SDK-Java

```
CoupageAds.INSTANCE.init(context,"affiliateId","subId");//If you have a subId, please add it.
```

Init SDK-Kotlin

```
CoupageAds.init(context, "affiliateId","subId");//If you have a subId, please add it.
```

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

layout

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

1.5. Load an ad (광고 로딩)

Call the `loadAdData ()` method to load the advertisement. Before the advertisement data is returned, the status of `AdsBannerView` must be `"isAttachedToWindow=true"`

광고 로딩을 위해 `loadAdData ()` method를 호출한다. 광고 데이터가 반환되기 전 `AdsBannerView` 상태는 `"isAttachedToWindow=true"`이어야 한다.

Load an ad-Java

```
adView=findViewById(R.id.ad_view);
adView.setAdSize(AdSize.BANNER_320X50);

adView.setAdListener(new AdListener() {
    @Override
    public void onAdLoaded() {
        // Code to be executed when an ad finishes loading.
    }

    @Override
    public void onAdFailedToLoad(@Nullable String errorMessage) {
        // Code to be executed when an ad request fails.
    }

    @Override
    public void onAdClicked() {
        // Code to be executed when the user clicks on an ad.
    }
});

adView.loadAdData(widgetId);
```

Load an ad-Kotlin

```
val adView = findViewById<AdsBannerView>(R.id.ad_view)
adView.setAdSize(AdSize.BANNER_320X50)
adView.setAdListener(object : AdListener {
    override fun onAdLoaded() {
        // Code to be executed when an ad finishes loading.
    }

    override fun onAdFailedToLoad(errorMessage: String?) {
        // Code to be executed when an ad request fails.
    }

    override fun onAdClicked() {
        // Code to be executed when the user clicks on an ad.
    }
})
adView.loadAdData(widgetId)
```

1.6. BannerView Size (배너 크기)

AdSize constant (AdSize 상수)	Size dp (WxH)
AdSize.BANNER_320X50	320x50
AdSize.LARGE_BANNER_320X100	320x100
AdSize.MEDIUM_RECTANGLE_300X250	300x250
AdSize.SMART_BANNER_WIDTH	Screen width (화면 너비) x 32 50 90 Screen width (화면 너비) <=400dp height (높이)=32 400dp<Screen width (화면 너비) <=720dp height (높이)=50 Screen width (화면 너비) >720dp height (높이)=90

1.7. 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" />
...
```

1.8. Load an ad (광고 로딩)

Call the `loadAdData ()` method to load the advertisement. Before the advertisement data is returned, the status of `AutoScrollBannerView` must be `"isAttachedToWindow=true"`

광고 로딩을 위해 loadAdData () method를 호출한다. 광고 데이터가 반환되기 전 AutoScrollBannerView 상태는 "isAttachedToWindow=true"이어야 한다.

Load an ad-Java

```
adView=findViewById(R.id.auto_scroll_banner);

adView.setAdListener(new AdListener() {
    @Override
    public void onAdLoaded() {
        // Code to be executed when an ad finishes loading.
    }

    @Override
    public void onAdFailedToLoad(@Nullable String errorMessage) {
        // Code to be executed when an ad request fails.
    }

    @Override
    public void onAdClicked() {
        // Code to be executed when the user clicks on an ad.
    }
});

adView.startAutoPlay() //Turn on automatic scrolling.default option
adView.stopAutoPlay() //Turn off automatic scrolling

adView.loadAdData(widgetId);
```

Load an ad-Kotlin

```
val adView = findViewById<AdsBannerView>(R.id.auto_scroll_banner)
adView.setAdListener(object : AdListener {
    override fun onAdLoaded() {
        // Code to be executed when an ad finishes loading.
    }

    override fun onAdFailedToLoad(errorMessage: String?) {
        // Code to be executed when an ad request fails.
    }

    override fun onAdClicked() {
        // Code to be executed when the user clicks on an ad.
    }
})

adView.startAutoPlay() //Turn on automatic scrolling.default option
adView.stopAutoPlay() //Turn off automatic scrolling

adView.loadAdData(widgetId)
```

1.9. LoadData param (loadData 파라미터)

You can choose to add the page name where the ad is located and the position of the ad in the page.

At present, it will not have any impact on the results.

광고가 노출되는 페이지의 페이지명과 페이지 상에 광고가 노출되는 위치명을 추가할 수 있다.

현재는 이렇게 해도 결과에 어떠한 영향도 주지 않는다.

param	
widgetId	you must pass in widgetId 위젯 ID 제출
affiliatePage	ad page name 광고 페이지명
affiliatePlacement	<i>the position of the ad in the page</i> 페이지 상 광고의 노출 위치

1.10. Ad events (광고 이벤트)

methods	
onAdLoaded()	Code to be executed when an ad finishes loading. 광고 로딩이 완료되었을 때 코드가 실행된다.
onAdFailedToLoad(errorMessage)	Code to be executed when an ad request fails.errorMessage is the error description. 에러 설명이 ad request fails.errorMessage일 경우 코드가 실행된다.
onAdClicked()	Code to be executed when the user clicks on an ad. 유저가 광고를 클릭하면 코드가 실행된다.