

Mobile Android SDK Usage Guide (모바일 안드로이드 SDK 사용 가이드) (v1.1.2) (KR)

- 1. 선행 요건 안내
- 2. 앱 설정 방법 안내
 - 2.1. 아래 옵션을 프로젝트별 'build.gradle' 파일에 추가하십시오.
 - 2.2. 모바일 광고 SDK의 경우 모듈 앱 단위 Gradle file (주로 "app/build.gradle") 에 dependency를 추가 하십시오.
 - 2.3. 릴리즈 시 해당 코드를 'proguard-rules.pro'에 설정하십시오.
- 3. AndroidManifest.xml 내 Affiliate ID & SubID 설정 안내
- 4. AdViewModel 생성 안내
 - 4.1. 세부 파라미터 설명
 - 4.1.1. Supported Creative Sizes (지원 가능 소재 사이즈)
- 5. 광고 데이터 로드 방법 안내
- 6. 광고 설정 및 표시 방법 안내
 - 6.1. 고정 배너
 - 6.1.1. 레이아웃에 AdsBannerView 를 추가하십시오.
 - 6.1.2. AdsBannerView에 AdViewModel 를 연동하십시오.
 - 6.2. 오토 스크롤 배너
 - 6.2.1. 레이아웃에 AutoScrollBannerView 를 추가하십시오.
 - 6.2.2. AdModel.SCROLL 파라미터를 통한 AdViewModel 를 생성하십시오.
 - 6.2.3. AutoScrollBannerView에 ViewModel 를 연동하십시오.
 - 6.3. 전면 광고 (Interstitial 광고)
 - 6.3.1. 광고 로드 시 Interstitial 객체를 생성하십시오.
 - 6.3.2. 전체 화면 광고(Interstitial 광고) 노출을 구현하십시오.
- 7. 에러 코드 안내
- 8. 안드로이드 13 업데이트에 따른 퍼미션 업데이트 안내
- 9. ads-log-plugin 디버그 진단 도구 설정 안내
- 10. 사용 방법 안내
 - 10.1.1. 타겟 API 레벨 23 이하의 환경에서의 쿠팡 SDK 연동 방법
 - 10.1.2. 배너 광고의 회색 테두리 제거 방법
 - 10.1.3. 배너 광고의 커스텀 기능 최대 활용 방법
 - 10.1.3.1. xml 파일(예: bg_banner.xml)을 앱의 res/drawable 폴더에 생성하십시오.
 - 10.1.3.2. 배너의 layout 파일에서 Android::background를 위 파일에 설정하십시오.

1. 선행 요건 안내

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

- Affiliate ID (required)/ 어필리에이트 ID(필수)
→ 쿠팡 파트너스 가입 아이디 (AF+7자리 숫자로 구성)
- Sub ID (required)/ 서브 ID(필수)
→ 8~20자 숫자 및 영어대소문자. Sub ID 등록 시 파트너스 내 "채널 아이디"에 등록된 정보와 반드시 매칭 되어야 하며, 최대 10개까지 등록 가능하다.
- Widget ID (required)/ 위젯 ID(필수)
→ 쿠팡 파트너스의 다이내믹 배너 생성 가이드를 참고하여 위젯 ID를 생성한다. 광고 노출을 위한 데이터 로딩을 위해서는 위젯 ID가 필수로 필요하다.

→ 위젯ID 배너 설정 옵션 : "배너타입 -> 고객 관심 기반 추천" / "배너 데이터 -> 추천 상품 피드" 로 등록하여 설정한다.

2. 앱 설정 방법 안내

2.1. 아래 옵션을 프로젝트별 'build.gradle' 파일에 추가하십시오.

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

만약 프로젝트가 settings.gradle 파일에 dependencyResolutionManagement를 정의한 경우(예: Android Studio에서 프로젝트 생성 마법사를 따라 새 Android 프로젝트를 생성한 경우), build.gradle 대신 이 파일의 URL을 declare 할 수 있습니다.

예시 >

```

dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        jcenter()

        // Repository for Coupang Ads:
        maven {url "https://raw.githubusercontent.com/coupang-ads-sdk/android/main"}
    }
}

```

2.2. 모바일 광고 SDK의 경우 모듈 앱 단위 **Gradle** file (주로 "*app/build.gradle*")에 **dependency**를 추가 하십시오.

```

dependencies {
    implementation 'com.coupang:ads:1.1.2'
}

```

2.3. 릴리즈 시 해당 코드를 'proguard-rules.pro'에 설정하십시오.

```

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

```

3. AndroidManifest.xml 내 Affiliate ID & SubID 설정 안내

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

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. AdViewModel 생성 안내

WidgetId, AdCreativeSize 및 optional 파라미터값을 통해 AdViewModel을 반영합니다.

- Kotlin 예시 :

Creative AdViewModel in kotlin

```

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

```

- Java 예시 :

```
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"
        );
    }
}
```

* 참고 : AdsViewModel은 Fragment class에서도 반영이 가능합니다.

4.1. 세부 파라미터 설명

AdsViewModel 작성에 하기 파라미터가 지원 됩니다:

param name(파라미터 명)	Description (세부 설명)	필수여부
widgetId	광고의 WidgetId, 상단의 '2. Prerequisites(선행요건)' 섹션 참조	예
adsCreativeSize	광고 소재 사이즈, 하단의 '5.1.1. Supported Creative Sizes (지원가능 소재 사이즈)' 섹션 참조	예
affiliatePage	앱 내 광고 페이지명	아니오
affiliatePlacement	페이지 상 광고의 노출 위치	아니오
adsMode	광고 display mode를 조정할 수 있는 Parameter 이며, Default 값은 AdsMod.AUTO. 입니다. 노트: Auto 스크롤 광고의 경우 해당 value를 AdMode.SCROLL 로 설정 해야 합니다.	아니오

4.1.1. Supported Creative Sizes (지원 가능 소재 사이즈)

AdsCreativeSize enum (CreativeSize 상수)	density-independent pixels(DP) 단위(WxH) 기준 사이즈
기존 일반 배너 사이즈: AdsCreativeSize._320X50 AdsCreativeSize._320X100 AdsCreativeSize._300X250 신규로 추가 된 사이즈: AdsCreativeSize._320X480 AdsCreativeSize._480X320 AdsCreativeSize._480X640 AdsCreativeSize._640X480 AdsCreativeSize._640X960 AdsCreativeSize._960X640	가로 x 세로 기준 (사이즈의 경우 각 AdsCreativeSize의 enum 함수값 참조)

AdsCreativeSize.SMART	<p>Screen width (화면 너비) x 32 50 90</p> <p>Screen width (화면 너비) <= 400 dp 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	<p>스마트폰 기기 내 320x480 (세로모드), 또는 480x320 (가로모드) 사이즈 지원</p> <p>태블릿 기기 내 640x960 (세로모드), 또는 960x640 (가로모드) 사이즈 지원</p> <p>해당 광고는 전체화면 중앙에 위치한 팝업 윈도우에 표시 됩니다.</p>

5. 광고 데이터 로드 방법 안내

Coupang에서 광고를 로드하려면 위에서 생성한 `adsViewModel` 기반의 `loadAdData()` 호출이 필요합니다. 또한 데이터 로드 결과 또는 발생 가능한 오류 메시지에 따라 아래 예시와 같이 작업을 할 수 있습니다. (하기 예시 참조) :

- Kotlin 예시:

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 예시:

LoadAdData with AdsViewModel in Java

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

// Observe ad loading results.
bannerAdsViewModel.observe(this, result -> {
    Result<Object> 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. 광고 설정 및 표시 방법 안내

고정 배너, 오토 스크롤 배너, Interstitial 광고(전면 광고)등 총 3가지 유형의 광고가 지원 되며, 각 형태에 따라 설정 및 구성이 상이합니다. (상세 내용 하단 참조)

6.1. 고정 배너

6.1.1. 레이아웃에 AdsBannerView 를 추가하십시오.

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. AdsBannerView에 AdViewModel 를 연동하십시오.

- Kotlin 예시:

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

6.2. 오토 스크롤 배너

6.2.1. 레이아웃에 AutoScrollBannerView 를 추가하십시오.

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. AdsModel.SCROLL 파라미터를 통한 AdViewModel 를 생성하십시오.

- Kotlin 예시:

Creative AdViewModel in kotlin

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

6.2.3. AutoScrollBannerView에 ViewModel 를 연동하십시오.

- Kotlin 예시:

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

6.3. 전면 광고 (Interstitial 광고)

Interstitial은 전체 화면 광고로, 화면 우측 상단 코너에 "닫기"버튼을 클릭 시에만 광고가 해제됩니다. 전면 광고 구현 방법은 다음과 같습니다 :

6.3.1. 광고 로드 시 Interstitial 객체를 생성하십시오.

- Kotlin 예시:

```
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. 전체 화면 광고(Interstitial 광고) 노출을 구현하십시오.

전체 화면 광고는 앱의 자연스러운 흐름 내 구현되어야 하며(예 : 게임 레벨 변경 시), 전면 광고를 노출하기 위해서는 showAds() 방식을 사용 하여야 합니다.

- Kotlin 예시:

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

7. 에러 코드 안내

에러 코드	해결 방법
Data is empty	이 오류는 Coupang 광고 서버가 광고 요청을 반환할 수 없을 경우 발생하며, 주로 하기 2가지 원인에서 발생 합니다 : 1) 광고 요청 디바이스가 어떤 Coupang user와도 맵핑되지 않음. 이 경우, 테스트를 진행한 동일 디바이스 내 Coupang App을 설치하고 로그인 후 재시도를 권장하며, Coupang App을 설치한 최초 시점부터 1일이 소요될 수 있는 관계로, 적어도 1일 후에 재시도 할 것을 추천 합니다. 2) 유저가 Coupang 내 Opt-out 되어 Coupang의 개인화 광고가 노출될 수 없음. 이 경우, Opt-out 되지 않은 계정으로 Coupang App에 로그인 후, 1일 후에 재시도 해보시기 바랍니다.
Ad id is empty or Failed to get Android ADID	이 오류는 android AdId(AAID)를 검색할 수 없을 경우 발생하며, 주로 하기의 2가지 원인에서 비롯될 가능성이 높습니다 : 1) 유저가 디바이스에서 개인화 광고를 Opt-out 지정하여 광고가 노출될 수 없음. 해당 디바이스에서 광고 노출 테스트를 원한다면 개인화 광고 노출 옵션을 Opt in 후 재시도 하십시오. 2) 간혹 유저가 App을 설치한지 얼마 지나지 않았거나 앱을 처음 사용했을 경우, 일시적으로 에러가 발생할 수 있음. 이 경우 에러는 최초 1회에만 발생하는 관계로, 해당 에러는 무시하십시오.

8. 안드로이드 13 업데이트에 따른 퍼미션 업데이트 안내

구글 정책 변화에 따라 Android 13 부터는 ADID 값 수집을 위해 추가적인 허가(Permission)값이 필요합니다. 이에 따라 현재 적용하고자 하는 앱이 Android 13(API 33) 적용 디바이스나 그 이상 버전을 타겟으로 하고 있다면 하단의 AndroidManifest.xml 를 추가하시기 바랍니다.

더 자세한 내용은 다음 링크를 참고 부탁드립니다: [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" />
```

9. ads-log-plugin 디버그 진단 도구 설정 안내

ads-log-plugin은 광고 SDK에서 생성된 logcat 로그를 파일로 내보내기 위한 진단 도구입니다. 해당 로그 데이터는 SDK 연동 중에 광고가 제대로 보여지지 않거나 충돌 등의 예외적인 오류가 발생했을 시, 정확한 이슈 원인 파악과 해결방안 검토에 매우 큰 도움이 될 수 있습니다.

해당 plugin은 하기 디펜던시를 프로젝트의 build.gradle 내 추가를 통해 쉽게 구현할 수 있습니다 :

```
dependencies {  
    implementation 'com.coupang:ads-log-plugin:1.0.0'  
}
```

만약 plugin을 release 버전이 아닌 debug 버전에만 추가하길 원하신다면, 하기 디펜던시를 적용하시기 바랍니다 :

```
dependencies {  
    debugImplementation 'com.coupang:ads-log-plugin:1.0.0'  
}
```

ads-log-plugin이 정상적으로 적용 되었다면, 타겟 디바이스의 "/data/data/\${ApplicationId}/files/adslog" 경로에서 아웃풋 로그 파일을 찾을 수 있습니다.

10. 사용 방법 안내

10.1.1. 타겟 API 레벨 23 이하의 환경에서의 쿠팡 SDK 연동 방법

쿠팡 광고 ADK는 안드로이드 API 23 이상의 버전이 필요합니다. 앱의 타겟 API 수준이 23 이상이 아닌 경우, 하기 두 가지 추가적인 방법을 통해 쿠팡 SDK와 연동할 수 있습니다 :

1) AndroidManifest.xml 에서, 하기와 같은 xml node 값을 <manifest>와 </manifest>블록 사이에 추가합니다:

```
<uses-sdk tools:overrideLibrary="com.coupang.ads,com.coupang.foundation,com.coupang.ads.clog" />
```

이렇게 하면 Coupang SDK로 앱을 빌드할 수 있습니다.

2) 런타임 실행 시, 디바이스의 실제 Android API 레벨이 23 이상(i.e. Android Version code M)인지 확인합니다:

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
    // Initialize and use SDK  
}
```

10.1.2. 배너 광고의 회색 테두리 제거 방법

쿠팡 배너 광고는 기본적으로 흰색 바탕에 회색 테두리가 설정되어 있습니다. 회색 프레임을 제거하려면 하기 예시와 같이 배너의 레이아웃 파일에 android:background 속성을 추가하십시오.

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

회색 테두리를 제거와 함께 배경색을 동시에 변경해야 할 경우, 위의 #FFFFFF 색상 코드 부분을 원하는 색깔로 변경 하십시오.

10.1.3. 배너 광고의 커스텀 기능 최대 활용 방법

쿠팡 배너 광고는 기본적으로 흰색 바탕에 회색 테두리가 설정되어 있습니다. 가능한 옵션을 최대 활용하여 커스텀 진행을 원하실 경우, 하기와 같이 단계 별 예시를 참고 하십시오.

유의사항 - 광고 배경 커스텀 기능의 경우, Interstitial 광고에서는 지원되지 않습니다.

10.1.3.1. xml 파일(예: bg_banner.xml)을 앱의 res/drawable 폴더에 생성하십시오.

이 파일은 광고 배경 설정 활용 방법에 대한 안내입니다. 예를 들어, 배너 배경색을 노란색으로 채우고, 테두리를 블루 그레이색으로 설정한 다음, 모서리를 둥글게 만드는 경우 하기 예시와 같이 구현이 가능합니다 :

```
<?xml version="1.0" encoding="utf-8"?>  
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">  
    <corners android:radius="4dp" />  
    <stroke android:width="1dp" android:color="@color/bluegray" />  
    <solid android:color="@color/yellow" />  
</shape>
```

10.1.3.2. 배너의 layout 파일에서 Android::background를 위 파일에 설정하십시오.

설정 예시 :

```
<com.coupang.ads.view.banner.AdsBannerView  
    android:id="@+id/ad_view"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@drawable/bg_banner" />
```

더 상세한 배경 사용자 지정을 원하는 경우 [Android Drawable Overview](#)를 참조하여, 위의 Android:background에서 drawable 소유자를 정의하십시오.