Mobile Android SDK Usage Guide (v1.1.0)

1. Introduction (목차)

2. 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 파라미터 값 설정이 필요합니다.

- **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 하며, 최대 10개까지 등록 가능하다.
- **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 필수로 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 등록하여 설정한다.

3. Configure your app (앱 설정)

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

아래 옵션을 프로젝트별 'build.gradle' 파일에 추가하십시오.

```
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 추가하십시오.

```
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 설정하십시오.

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

4. Config Affiliate ID & Sub ID in AndroidManifest.xml (AndroidManifest.xml 내 Affiliate ID & SubID 설정)

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

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>
```

5. Create AdsViewModel [광고 View 형태(AdsViewModel) 생성하기]

Create an AdsViewModel with widgetId, AdsCreativeSize and other optional parameters.

WidgetId, AdsCreativeSize 및 optional 파라미터값을 통해 AdsViewModel을 반영합니다.

- **Kotlin example (Kotlin 예시) :**

Creative AdsViewModel in kotlin

```
class KtActivity: AppCompatActivity() {
    ...
    private val bannerAdsViewModel: AdsViewModel by createAdsViewModel(
            "514017",  // your widgetId
            AdsCreativeSize._320x100,
                affiliatePage = "MainPage",
                affiliatePlacement = "Bottom Banner"
    )
    ...
}
```

- **Java example (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"
                );
        }
}
```

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

*** 참고 : AdsViewModel은 Fragment class에서도 반영이 가능합니다.**

5.1. Parameters description (세부 파라미터 설명)

The creation of AdsViewModel supports the following parameters :

AdsViewModel 작성에 하기 파라미터가 지원 됩니다:

param name(파라미터)	Description (세부 설명)	Required?
`widgetId`	**WidgetId for the ad. See the Prerequisites section above for details.** **광고의 WidgetId, 상단의 '2. Prerequisites(선행요건)' 섹션 참조**	**Yes**
adsCreativeSize	**Size of the ad. See the 'Supported Creative Sizes' section below for details** **광고 소재 사이즈, 하단의 '5.1.1. Supported Creative Sizes (지원가능 소재 사이즈)' 섹션 참조**	**Yes**
`affiliatePage`	**The page name in the app** **앱 내 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 *AdsMode.AUTO*.** **Note: For auto scroll ads, set this value to *AdsMode.SCROLL*** **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.**	**No**

5.1.1. Supported Creative Sizes (지원 가능 소재 사이즈)

AdsCreativeSize enum (CreativeSize 상수)	Size in DP unit (WxH)
`AdsCreativeSize._320X50`	**320x50**
`AdsCreativeSize._320X100`	**320x100**
`AdsCreativeSize._300X250`	**300x250**
`AdsCreativeSize.SMART`	***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**
AdsCreativeSize.INTERSTITIAL	***Screen Width (화면 너비)* x *Screen Height (화면 높이)* (i.e. This is a full screen ad / 참고_해당 사이즈는 전체 화면 광고입니다.)**

6. Load Ads Data (광고 데이터 로드)

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.

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 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<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();
```

7. Configure and display the Ads (광고 설정 및 표시)

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

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)

7.1. Static Banner (고정 배너)

7.1.1. 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" />
...
```

7.1.2. Bind AdsViewModel to AdsBannerView (AdsBannerView에 AdsViewModel 연동)

- **Kotlin example:**

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

7.2. Auto Scroll Banner (오토 스크롤 배너)

7.2.1. 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" />
...
```

7.2.2. Create AdsViewModel with AdsModel.SCROLL parameter (AdsModel.SCROLL 파라미터를 통한 AdsViewModel 생성)

- **Kotlin example:**

Creative AdsViewModel in kotlin

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

7.2.3. Bind ViewModel to AutoScrollBannerView (AutoScrollBannerView에 ViewModel 연동)

- **Kotlin example:**

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

7.3. Interstitial Ads (전면 광고)

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 :

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

7.3.1. Create an Interstitial object when Ad loaded. (광고 로드 시 Interstitial 객체 생성)

- **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()
    }
}
```

7.3.2. Show the interstitial ad. (전체 화면 광고 노출)

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.

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

- **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.")
                }
        }
...
}
```

8. Error codes (에러 코드)

Error codes 에러 코드	Solution 해결 방법
Data is empty	This error means Coupang ads server didn't return any ad for the request. This usually happens in two cases : 이 오류는 Coupang 광고 서버가 광고 요청을 반환할 수 없을 경우 발생하며, 주로 하기 2가지 원인에서 발생 합니다 : 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. 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) 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. 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	This error means the android AdId for advertising (a.k.a AAID) can't be retrieved. This might happen in two cases: 이 오류는 android AdId(AAID)를 검색할 수 없을 경우 발생하며, 주로 하기의 2가지 원인에서 비롯될 가능성이 높습니다 : 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. 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) 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). 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.

9. 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()

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 버젼을 타깃으로 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()

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

10. Debug tools ads-log-plugin (ads-log-plugin 디버그 진단 도구 설정 안내)

ads-log-plugin is a diagnostic tool for exporting logcat logs generated by Ads SDK to files. When you see exceptions from the SDK during integration, for example, crashed, ads not displayed properly, etc. These logs will be very helpful for us to troubleshoot the issue and provide solution to unblock your integration.

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.

The plugin is very easy to integrate, you just need to add this dependency to your project's build.grade:

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

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

If you only want to include the plugin in debug version but not the release version, you can configure the dependencies as:

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

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

When ads-log-plugin is working properly, you can find the output log files under the target device "/data/data/${ApplicationId}/files/adslog" path.

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