

# Cougang Ads iOS SDK User Guide (v1.0.0)

- 1. Prerequisites (선행 요건 안내)
- 2. Development Environment Requirements (개발 환경 요구사항 안내)
- 3. Configure your app (앱 설정 안내)
  - 3.1. Import the Mobile Ads SDK
  - 3.2. Embed & Sign the CouOffsiteAds.xcframework
  - 3.3. Update your Info.plist
- 4. Initialize the Mobile Ads SDK
  - 4.1.1. Example code
- 5. Banner Ads
  - 5.1. Create a BannerView
    - 5.1.1. Example to create CouAdBannerView Programmatically
    - 5.1.2. Parameters description (세부 파라미터 설명)
    - 5.1.3. Supported Creative Sizes (지원 가능 소재 사이즈)
  - 5.2. Listen to Ad Events
    - 5.2.1. Registering for banner ad events
    - 5.2.2. Implementing banner ad events

## 1. Prerequisites (선행 요건 안내)

To use the Cougang Ads iOS SDK, an account need to be registered on Cougang Partners (<https://partners.cougang.com/#help/partners-guide>), and the following parameters need to be available:  
쿠팡 광고 SDK 사용을 위해서는 쿠팡 파트너스 (<https://partners.cougang.com/#help/partners-guide>) 에 계정을 생성, 등록해야 하며, 아래 파라미터 값 설정이 필요합니다.

- Affiliate ID (required)/ 어필리에이트 ID(필수)
  - Cougang 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 Cougang 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.

## 2. Development Environment Requirements (개발 환경 요구사항 안내)

- Xcode: Version 14.0 or higher
- Target iOS: Version 12.0 or higher

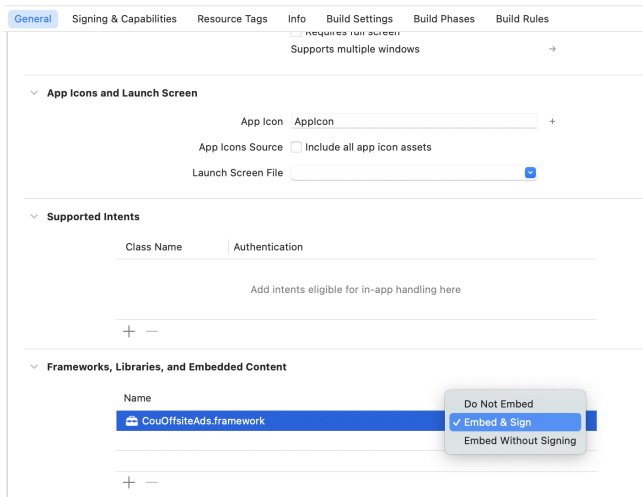
## 3. Configure your app (앱 설정 안내)

### 3.1. Import the Mobile Ads SDK

Download the [SDK framework](#) directly, and import the `CouOffsiteAds.xcframework` into your Xcode project.  
SDK 프레임워크를 직접 다운로드하여, `CouOffsiteAds.xcframework` 프레임워크를 Xcode 프로젝트에 임포트합니다.

### 3.2. Embed & Sign the CouOffsiteAds.xcframework

- General -> Frameworks, Libraries, and Embedd Content -> `CouOffsiteAds.xcframework` -> Embed & Sign



### 3.3. Update your Info.plist

Update your app's `Info.plist` file to add `NSUserTrackingUsageDescription` keys.  
앱의 `Info.plist` 파일을 업데이트하여 `NSUserTrackingUsageDescription` 키를 추가합니다.

## 4. Initialize the Mobile Ads SDK

Before loading ads, call the `startWith(affiliateId: "", subId: "")` method on the `CouAdsManager.shared`, which initializes the SDK with your affiliate ID and subId (please refer to the 1. Prerequisites section for details). This only needs to be done once, ideally at app launch time.

광고를 로드하기 전에 `CouAdsManager.shared`에서 `startWith(affiliateId: "", subId: "")` 메소드를 호출하여 affiliate ID 및 subId로 SDK를 설정합니다 (자세한 내용은 1. 선행요건 섹션을 참조하세요). 이 작업은 앱 실행 시 한 번만 수행하면 됩니다.

#### 4.1.1. Example code

```
import CouOffsiteAds

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplicationLaunchOptionsKey: Any]?) -> Bool {

        CouAdsManager.shared.startWith(affiliateId: "AF0123456", subId: "BannerAdsDemo")

        return true
    }
}
```

## 5. Banner Ads

Banner ads occupy a spot within an app's layout, either at the top or bottom of the device screen. They stay on screen while users are interacting with the app. If you're new to mobile advertising, they're a great place to start with.

배너 광고는 기기 화면의 상단 또는 하단에서 앱 레이아웃의 한 자리를 차지합니다. 사용자가 앱과 상호 작용하는 동안 화면에 유지됩니다. 모바일 광고가 처음이라면 배너 광고로 시작하기에 좋습니다.

### 5.1. Create a BannerView

Banner ads are displayed in `CouAdBannerView` class, so the first step toward integrating banner ads is to include a `CouAdBannerView` in your view hierarchy. This is typically done either with the Interface Builder or programmatically.  
배너 광고는 `CouAdBannerView` 클래스에 표시되므로 배너 광고를 통합하기 위한 첫 번째 단계는 뷰 계층 구조에 `CouAdBannerView`를 포함하는 것입니다. 이 작업은 일반적으로 인터페이스 빌더를 사용하거나 프로그래밍 방식으로 수행합니다.

### 5.1.1. Example to create `CouAdBannerView` Programmatically

A `CouAdBannerView` is instantiated directly. Here's an example of how to create a `CouAdBannerView`, aligned to the bottom center of the safe area of the screen, with a banner size of 320x50:

`CouAdBannerView`는 직접 인스턴스화됩니다. 다음은 화면의 안전 영역 하단 중앙에 정렬되고 배너 크기가 320x50인 `CouAdBannerView`를 만드는 방법의 예시입니다:

```
import import CouOffsiteAds
import UIKit

class ViewController: UIViewController {

    var bannerView: CouAdBannerView!

    override func viewDidLoad() {
        super.viewDidLoad()

        // In this case, we instantiate the banner with desired ad size.
        createFixedBanner50()
    }

    func createFixedBanner50() {
        let banner = CouAdBannerView(
            widgetId: "514017", // your widgetId
            bannerType: .FIXED_50,
            affiliatePlacement: "Bottom Banner",
            affiliatePage: "MainPage"
        )

        self.view.addSubview(banner)
        banner.delegate = self

        NSLayoutConstraint.activate([
            banner.topAnchor.constraint(equalTo: self.view.topAnchor, constant: 100),
            banner.centerXAnchor.constraint(equalTo: self.view.centerXAnchor)
        ])
    }
}
```

Note that in this case we don't give width or height constraints, as the provided ad size will give the banner an intrinsic content size. 이 경우 제공된 광고 크기에 따라 배너에 고유한 콘텐츠 크기가 제공되므로 너비 또는 높이 제한을 두지 않습니다.

### 5.1.2. Parameters description (세부 파라미터 설명)

The creation of `CouAdBannerView` supports the following parameters :  
`CouAdBannerView` 생성은 다음 파라미터를 지원합니다:

param name(파라미터 )	Description (세부 설명)	Required?
widgetId	WidgetId for the ad. See the Prerequisites section above for details. 광고의 WidgetId, 상단의 '1. Prerequisites(선행요건)' 섹션 참조	Yes
bannerType	Size of the ad. See the '5.1.3 Supported Creative Sizes' section below for details 광고 소재 사이즈, 하단의 '5.1.3. 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

### 5.1.3. Supported Creative Sizes (지원 가능 소재 사이즈)

AdsCreativeSize enum (CreativeSize 상수)	Size in density-independent pixels (DP) unit (WxH)
CouAdBannerType.SMART	<i>Screen width (화면 너비) x 32 50</i> <i>Screen width (화면 너비) &lt;= 375 dp height (높이)=32</i> <i>Screen width (화면 너비) &gt;375 dp height (높이)=50</i>
CouAdBannerType.FIXED_50	<i>Fixed size 320 x 50</i>
CouAdBannerType.FIXED_100	<i>Fixed size 320 x 100</i>
CouAdBannerType.FIXED_250	<i>Fixed size 320 x 250</i>

## 5.2. Listen to Ad Events

Through the use of `CouAdBannerViewDelegate`, you can listen for lifecycle events, such as when an ad is closed or the user leaves the app. `CouAdBannerViewDelegate` 사용하면 광고가 종료되거나 사용자가 앱을 떠날 때와 같은 라이프사이클 이벤트를 수신할 수 있습니다.

### 5.2.1. Registering for banner ad events

To register for banner ad events, set the `delegate` property on `CouAdBannerView` to an object that implements the `CouAdBannerViewDelegate` protocol. Generally, the class that implements banner ads also acts as the delegate class, in which case, the `delegate` property can be set to `self`.

배너 광고 이벤트를 등록하려면 `CouAdBannerView`의 델리게이트 속성을 `CouAdBannerViewDelegate` 프로토콜을 구현하는 객체로 설정합니다. 일반적으로 배너 광고를 구현하는 클래스가 델리게이트 클래스 역할도 하며, 이 경우 델리게이트 속성을 `self`로 설정할 수 있습니다.

```
import CouOffsiteAds
import UIKit

class ViewController: UIViewController, CouAdBannerViewDelegate {

    var bannerView: CouAdBannerView!

    override func viewDidLoad() {
        super.viewDidLoad()
        ...
        bannerView.delegate = self
    }
}
```

### 5.2.2. Implementing banner ad events

This example implements each method and logs a message to the console: 각 메소드를 구현하고 콘솔에 메시지를 기록하는 예시입니다:

```
extension ViewController: CouAdBannerViewDelegate{

    func bannerDidClick(_ bannerView: CouAdBannerView) {
        print("\(bannerView) banner did click")
    }

    func bannerViewFailToReceiveAdWithError(_ bannerView: CouAdBannerView, error: Error) {
        print("\(bannerView) fail to receive ad data")
    }

    func bannerDidReceiveAd(_ bannerView: CouAdBannerView) {
        print("\(bannerView) banner did receive ad data")
    }

    func bannerDidUpdateData(_ bannerView: CouAdBannerView) {
        print("\(bannerView) banner did update data")
    }
}
```