



# AD@M-IOS PUBLISHER SDK GUIDE

---

이 문서는 DAUM 신디케이션 제휴 당사자에 한해 제공되는 자료로 가이드 라인을 포함한 모든 자료의 지적 재산권은 주식회사 다음커뮤니케이션이 보유하고 있습니다.

Ver 2.2.2.1 .....	2
변경이력 .....	3
Ad@m 광고 삽입 방법 .....	7
개발 환경 .....	7
Ad@m SDK 구성 .....	7
1단계 : Client ID 발급받기 및 SDK 다운로드 .....	7
2단계 : 라이브러리 Import .....	8
3단계 : Build Settings .....	9
4단계 : 광고 뷰 삽입 및 호출 .....	10
a. 프로그램적인 삽입 방법 .....	10
B. 인터페이스 빌더를 이용한 삽입 방법 .....	11
5단계 : 전면형(Interstitial) 광고 호출 .....	13
Class Reference .....	14
AdamAdView .....	14
광고 뷰 생성 메소드 .....	14
광고 요청 / 요청 중단 메소드 .....	14
Properties .....	15
Constants .....	18
AdamAdViewDelegate .....	19
광고 수신 / 실패 관련 delegate 메소드 .....	19
광고 클릭 액션 관련 delegate 메소드 .....	20
AdamInterstitial .....	22
객체 생성 메소드 .....	22
광고 요청 메소드 .....	22
Properties .....	23
AdamInterstitialDelegate .....	25
광고 수신 / 실패 관련 delegate 메소드 .....	25
광고 노출 관련 delegate 메소드 .....	25
AdamError .....	28
Constants .....	28
추가 정보 .....	29
광고 수신 실패 원인에 따른 처리 방법 .....	29

## VER 2.2.2.1

이 가이드는 iOS Application에 모바일 광고를 노출하기 위한 광고 데이터요청과 처리 방법을 설명합니다.

사이트/앱 운영정책에 어긋나는 경우 적립금 지급이 거절 될 수 있으니 유의하시기 바랍니다.

- 문의 고객센터 <http://cs.daum.net/mail/form/256.html>
- 사이트/앱 운영 정책 [http://mobile.biz.daum.net/guide/guide\\_siteapp\\_policy.jsp](http://mobile.biz.daum.net/guide/guide_siteapp_policy.jsp)

- Ver 1.1a
  - ✓ WebView에서 고정값으로 계산되어 잘못 나타나던 부분에 대한 수정
- Ver 1.1b
  - ✓ 동영상광고에 대한 처리 수정
  - ✓ 잠정적 문제가 될 수 있는 메모리 누수부분 수정
- ver1.2
  - ✓ AD@m 클릭스 광고 지원
  - ✓ Xcode 4.x 부터는 다음 framework 들을 추가 필요
    - libxml2.2.dylib
    - libxml2.2.7.3.dylib
- ver1.3
  - ✓ 위치정보 등 수집 항목 제거(location, gender, birthday)
  - ✓ textcolor, backgroundcolor변경 지원 안함
  - ✓ 광고 library인 libAdAtDaum.a를 하나로 합침
  - ✓ MobileAdViewDelegate의 필수적이지 않은 함수들 optional로 수정
    - testDevice
    - didReceiveRefreshAd
    - didFailedReceiveAdWithError:
    - didFailedReceiveRefreshAdWithError:
    - willShowADInFullScreen
    - closedFullScreenAD
    - willResideAd
  - ✓ didFailedRecieveAdWithError -> didFailedReceiveAdWithError 스펠링 오류 수정
- ver1.3.1
  - ✓ 광고 클릭 후 webview가 보여진 상태에서 background/foreground 전환 시 crash 발생하는 문제 수정

- **ver1.3.2**
  - ✓ iOS5 대응
  - ✓ 간헐적으로 발생하는 XML Parsing Error 수정
  - ✓ 샘플코드에서 불필요 라이브러리 제거 (libxml2.x)
  - ✓ ANIMATIONTYPE\_NONE 추가
  
- **ver1.4.0**
  - ✓ 다양한 광고 Targeting 기능 추가(위치, 망, 통신사 등)
  - ✓ 추가 framework 필요
    - CoreLocation.framework
    - CoreTelephony.framework
  
- **ver1.4.1**
  - ✓ 특정 타입의 광고 노출시 간헐적으로 Crash 발생하는 문제 수정
  - ✓ 특정 타입의 광고 노출시 Status bar를 이용한 최상단 스크롤 안되는 문제 수정
  - ✓ 광고 페이지 로드시 콘솔에 에러 로그 출력되는 현상 수정
  
- **ver1.5.0**
  - ✓ SDK 구조 및 API 개선
    - 광고 뷰 클래스 변경: MobileAdView → AdamAdView
    - 광고 뷰에 Singleton 구조 적용
    - 광고 뷰 설정 및 상태 확인은 모두 property를 이용하도록 변경
    - 모든 delegate 메소드는 optional로 변경
    - delegate 객체로 UIViewController 객체를 지정하지 않으면 오류 생기는 문제 수정
    - 광고 자동 요청을 위한 내장 메소드 추가
    - 광고 수신 실패에 대한 에러 타입 세분화
  - ✓ ARC 기반의 일부 프로젝트에서 Crash 발생하는 현상 수정
  - ✓ 광고 뷰 Flexible Width 적용

- **ver2.0.0**
  - ✓ 리치미디어 광고 노출 기능 탑재
  - ✓ 전면형 광고 노출을 위한 AdamInterstitial 클래스 / AdamInterstitialDelegate 프로토콜 추가
  - ✓ AdamAdViewDelegate 프로토콜에 메소드 추가
    - didOpenFullScreenAd:
    - didCloseFullScreenAd:
  - ✓ 추가 Framework 필요
    - QuartzCore.framework
    - MediaPlayer.framework
- **ver2.1.0**
  - ✓ iOS 6 대응 (armv7s 아키텍처)
  - ✓ 권장 개발환경 변경
  - ✓ ASIHTTPRequest, Reachability 라이브러리 링크 에러 문제 수정
  - ✓ AdamAdView 프로퍼티 추가
    - birth, gender, sdkVersion
  - ✓ AdamInterstitial 프로퍼티 추가
    - birth, gender, sdkVersion
- **ver2.1.0.1**
  - ✓ ASIHTTPRequest, Reachability 라이브러리 사용 시 추가 문제 수정
- **ver2.2.0**
  - ✓ 리치미디어 광고 노출 추가 기능 탑재
  - ✓ 추가 Framework 필요
    - EventKit.framework
    - EventKitUI.framework
    - AssetsLibrary.framework
  - ✓ 추가 Linker Flag 필요: -ObjC
- **ver2.2.1**
  - ✓ iOS 7 대응
  - ✓ 추가 Framework 필요
    - AdSupport.framework

- ver2.2.2
  - ✓ iOS 7 추가 대응 및 UI 개선
  - ✓ Bug Fix
- ver2.2.2.1
  - ✓ 64bit 대응

## Ad@m 광고 삽입 방법

### 개발 환경

---

- ✓ IDE: Xcode 5.0 이상 권장
- ✓ Base SDK: iOS 7.0 이상
- ✓ iOS Deployment Target: iOS 4.3 이상

### Ad@m SDK 구성

---

- ✓ libAdamPublisher.a : Ad@m 광고 라이브러리 파일
- ✓ AdamAdView.h : 라이브러리 사용을 위해 필요한 광고 뷰 클래스 헤더
- ✓ AdamInterstitial.h : 라이브러리 사용을 위해 필요한 interstitial 클래스 헤더
- ✓ AdamError.h: 라이브러리에서 공통으로 사용되는 Error 클래스 헤더
- ✓ AdamAdWrapperView.h : 인터페이스 빌더에서 사용하기 위한 Wrapper 클래스 헤더
- ✓ AdamAdWrapperView.m : 인터페이스 빌더에서 사용하기 위한 Wrapper 클래스 소스
- ✓ Sample : Ad@m SDK를 적용한 샘플 프로젝트

### 1단계 : Client ID 발급받기 및 SDK 다운로드

---

실제 광고를 수신하고, 수익 창출을 위해서는 mobile.biz.daum.net에서 사이트/앱 등록 후 Client ID를 발급받아야 한다. 아래 URL을 통해 애플리케이션을 등록할 수 있다.

[http://mobile.biz.daum.net/guide/guide\\_siteapp1.jsp](http://mobile.biz.daum.net/guide/guide_siteapp1.jsp)

애플리케이션 등록 후 SDK를 다운로드 받는다.

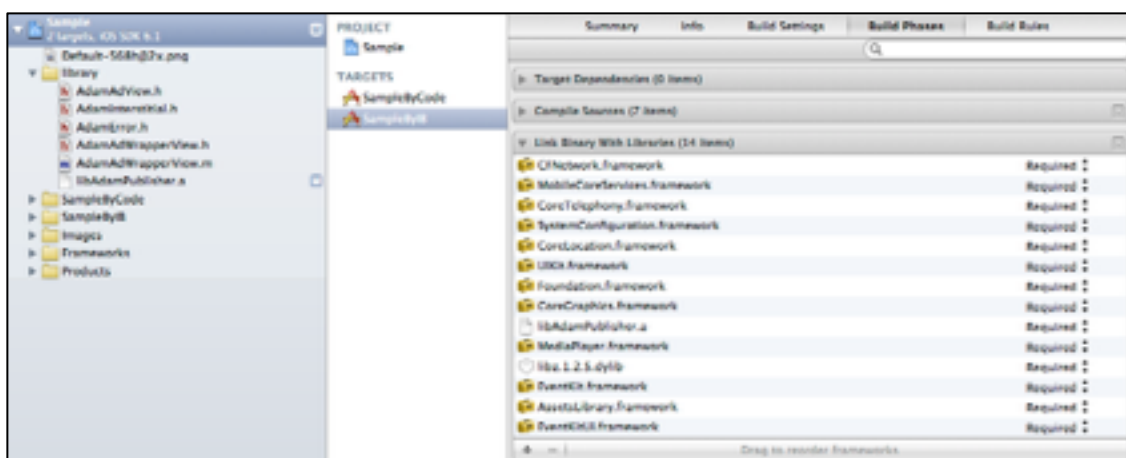
## 2단계 : 라이브러리 Import

다운로드 받은 SDK에서 library에 해당하는 폴더를 Xcode의 프로젝트 내에 복사한다.  
만약 인터페이스 빌더를 이용해 광고 뷰를 적용할 것이 아니라면 AdamAdWrapperView.h,  
AdamAdWrapperView.m 파일은 제외하여도 된다.

다음과 같은 framework가 없다면 추가로 포함시킨다.

2.2 버전 부터는 EventKit.framework, EventKitUI.framework, AssetsLibrary.framework이 추가적으로 필요하다. 또한, QuartzCore.framework은 더이상 사용하지 않으므로 제거해도 무방하다.

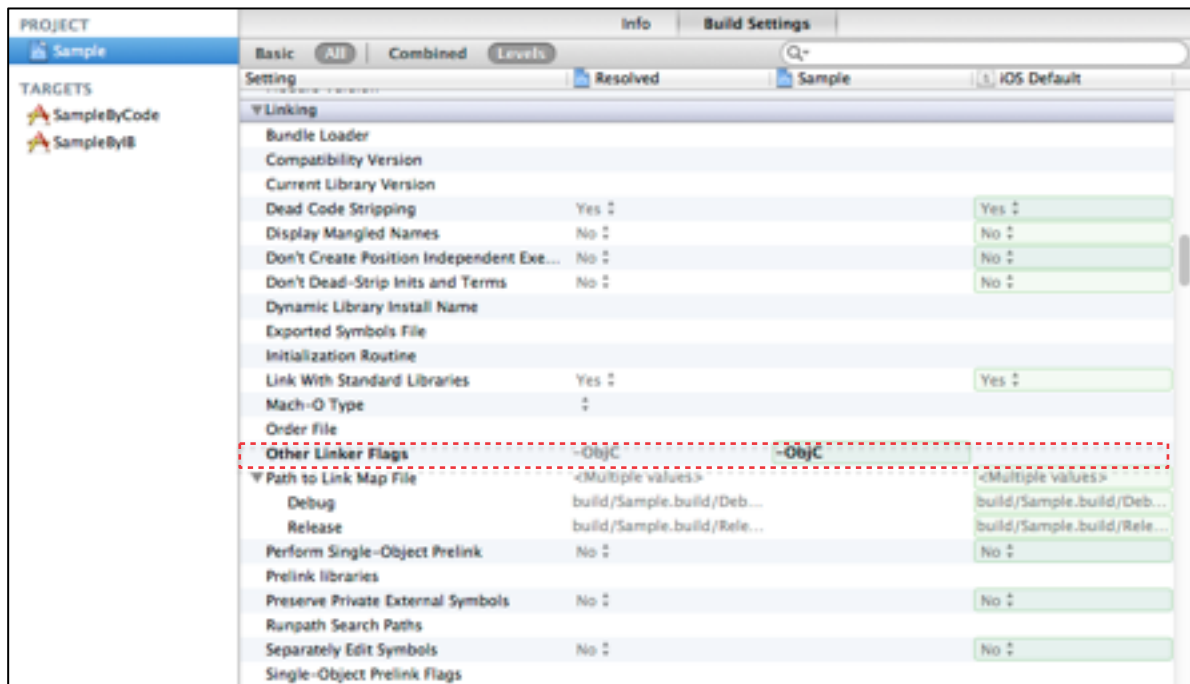
- ✓ UIKit.framework
- ✓ Foundation.framework
- ✓ CoreGraphics.framework
- ✓ MobileCoreServices.framework
- ✓ CFNetwork.framework
- ✓ SystemConfiguration.framework
- ✓ CoreLocation.framework
- ✓ CoreTelephony.framework
- ✓ MediaPlayer.framework
- ✓ EventKit.framework
- ✓ EventKitUI.framework
- ✓ AssetsLibrary.framework
- ✓ AdSupport.framework
- ✓ libz.1.2.5.dylib





### 3단계 : Build Settings

2.2 버전 부터는 정상적으로 SDK를 사용하기 위해 Build Settings > Linking > Other Linker Flags 항목에 -ObjC 플래그를 추가해주어야 한다.



## 4단계 : 광고 뷰 삽입 및 호출

### a. 프로그램적인 삽입 방법

code 작성을 통해 광고 뷰를 붙이는 방법은 다음의 순서를 따른다.

- ✓ AdamAdView.h 파일 import 하기
- ✓ 광고 뷰 생성하기
- ✓ 광고 뷰 객체에 Client ID 세팅하기
- ✓ 화면에 광고 뷰 붙이기

AdamAdView 클래스는 Singleton으로 구현되어있어, 하나의 애플리케이션에서 한개의 객체만 사용이 가능하다. 그렇기 때문에 애플리케이션 내의 여러 화면에서 광고를 보여주고자 하는 경우 가장 좋은 방법은 뷰 컨트롤러의 viewWillAppear: 또는 viewDidAppear: 메소드 내부에서 광고 뷰를 붙이는 것이다. 만약 하나의 화면에서만 광고를 보여주고자 한다면, viewDidLoad 메소드 내부에서 광고 뷰를 붙여도 무방하다.

```
// ViewController.m

#import "ViewController.h"
#import "AdamAdView.h"

@implementation ViewController

- (void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];

    // AdamAdView 객체를 가져온다.
    AdamAdView *adView = [AdamAdView sharedAdView];

    if (![adView.superview isEqual:self.view]) {
        // adView가 self.view에 붙어있는 상태가 아니라면,
        // adView에 필요한 속성을 설정한 후 self.view에 붙인다.
        adView.frame = CGRectMake(0.0, 0.0, 320.0, 48.0);
        adView.clientId = @"TestClientId";
        [self.view addSubview:adView];

        if (!adView.usingAutoRequest) {
            // adView가 광고 자동요청 기능을 사용하는 상태가 아니라면,
            // 60초 간격으로 광고 자동요청을 시작한다.
            [adView startAutoRequestAd:60.0];
        }
    }
}
```

## B. 인터페이스 빌더를 이용한 삽입 방법

인터페이스 빌더를 이용해 광고 뷰를 붙이는 방법은 다음의 순서를 따른다.

이 때에는 반드시 AdamAdWrappperView.h, AdamAdWrapperView.m 파일을 프로젝트에 추가해야 한다.

- ✓ AdamAdWrapperView.h, AdamAdWrapperView.m 파일 import 하기
- ✓ AdamAdWrapperView.m 파일에서 광고 뷰 설정하기
- ✓ 광고 뷰를 붙일 xib 파일에 새로운 뷰 추가하기
- ✓ 새로 추가한 뷰의 클래스를 AdamAdWrapperView로 지정하기

AdamAdWrapperView.m 파일을 열면 displayAdView라는 메소드가 구현되어있다.

이 메소드는 AdamAdWrapperView 객체가 화면에 보여질 때마다 호출되는데, 이 내부에 광고 뷰를 생성하여 붙이는 코드가 작성되어있다. 기 작성된 코드에서 clientId 속성만 자신의 실제 Client ID를 넣어주면 된다. 그 외의 속성에 대한 설정은 문서 뒷부분의 레퍼런스를 참고하여 추가하도록 한다.

```
// AdamAdWrapperView.m

#import "AdamAdWrapperView.h"
#import "AdamAdView.h"

@implementation ViewController

- (void)displayAdView
{
    AdamAdView *adView = [AdamAdView sharedAdView];

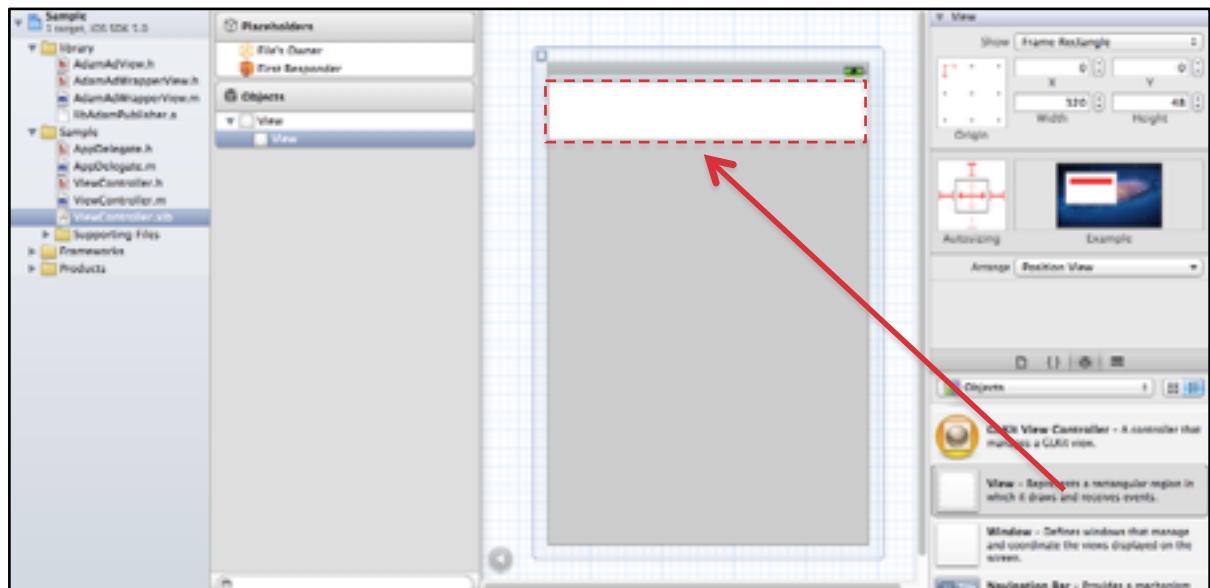
    if (![adView.superview isEqual:self]) {
        adView.frame = self.bounds;
        adView.autoresizingMask = UIViewAutoresizingFlexibleWidth;
        adView.clientId = @"TestClientId";
        [self addSubview:adView];

        if (!adView.usingAutoRequest) {
            [adView startAutoRequestAd:60.0];
        }
    }
}

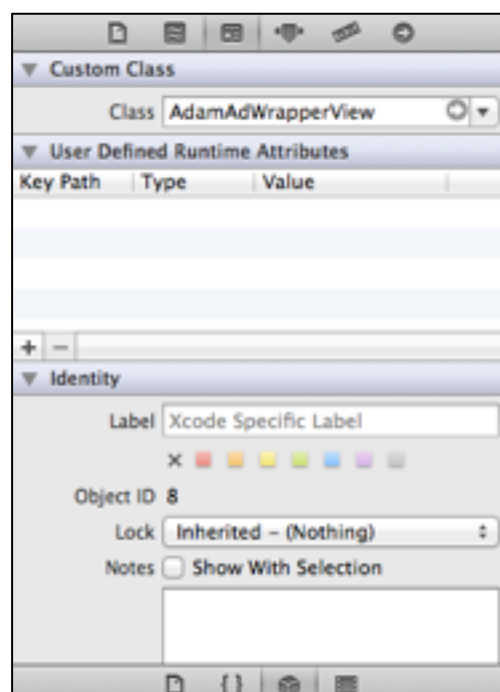
- (void)didMoveToWindow
{
    [super didMoveToWindow];

    if (self.window) {
        [self displayAdView];
    }
}
```

그 다음에는 광고 뷰를 붙일 뷰 컨트롤러의 xib 파일을 열고, View 요소를 추가한다.  
 뷰의 크기는 width: 320px, height: 48px로 설정하면 되고, 애플리케이션의 가로 모드 지원 여부를 고려하여  
 Autoresizing 항목을 적절히 설정한다.



마지막으로, 추가한 View 요소의 class를 지정한다. 화면 우측의 Identity inspector 창을 열고, class 입력창에  
 AdamAdWrapperView라고 입력해준다.



## 5단계 : 전면형(Interstitial) 광고 호출

Interstitial(전면형) 광고는 당분간 Ad@m의 네트워크 파트너를 대상으로 노출됩니다.

Ad@m의 네트워크 파트너가 아닐 경우에도 Expandable(확장형), Animated Banner (애니메이션형)의 Rich Media 광고가 노출됩니다.

전면형 광고를 호출하는 방법은 다음의 순서를 따른다.

이 때, 반드시 AdamInterstitial.h 파일을 프로젝트에 포함시켜야 한다.

- ✓ AdamInterstitial.h 파일 import 하기
- ✓ AdamInterstitial 객체 생성하기
- ✓ AdamInterstitial 객체에 clientId 세팅하기
- ✓ 전면형 광고 호출하기

AdamInterstitial 클래스는 전면형 광고를 호출하기 위해 사용하며, Singleton으로 구현되어있다.

AdamInterstitial 객체에 필요한 속성을 세팅해준 후 requestAndPresent 메소드를 호출하면 광고를 요청하게 되며, 성공적으로 광고를 수신한 경우 즉시 전체화면에 광고 화면을 노출한다.

광고를 호출하는 시점에는 제한이 없으나, 사용자가 애플리케이션을 사용하는데 큰 지장을 초래하지 않도록 배려해야 한다. (예: 게임 플레이 도중 전면형 광고 노출하는 경우)

전면형 광고의 최소 재호출 간격은 3분이며, 수신 성공/실패 여부에 관계없이 3분간은 전면형 광고를 다시 호출 할 수 없다.

```
// ViewController.m

#import "ViewController.h"
#import "AdamInterstitial.h"

@implementation ViewController

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];

    // AdamInterstitial 객체를 가져온다.
    AdamInterstitial *interstitial = [AdamInterstitial sharedInterstitial];

    // interstitial에 필요한 속성을 설정한 후 광고를 호출한다.
    interstitial.clientId = @"InterstitialTestClientId";
    [interstitial requestAndPresent];
}
```

## Class Reference

### AdamAdView

---

광고 뷰 자체를 가리키는 클래스로, 사용자는 헤더에 선언된 메소드 및 프로퍼티들을 이용해 광고 요청을 실행하고 광고 뷰의 동작을 설정할 수 있다.

AdamAdView 클래스의 객체를 생성하기 위해서는 아래에서 설명하는 `sharedAdView` 메소드만을 사용해야 하고, `alloc` 메소드는 사용할 수 없다. 또한 AdamAdView 클래스를 상속받는 서브클래스를 생성하여 사용하는 것도 불가능하다.

### 광고 뷰 생성 메소드

---

- **sharedAdView**

+ (AdamAdView \*)sharedAdView

AdamAdView 클래스의 Singleton 객체인 `sharedAdView`를 리턴한다.

AdamAdView 클래스는 Singleton으로 구현되었으므로, 하나의 앱에서는 하나의 AdamAdView 객체만을 사용할 수 있다.

#### Return

AdamAdView 객체

### 광고 요청 / 요청 중단 메소드

---

- **requestAd**

- (void)requestAd

광고를 1회 요청한다.

기존과 같이 타이머를 이용해 직접 광고를 요청하고자 할 때 사용할 수 있다.

최소 호출 가능한 시간 간격은 12초이고, 그 이내에 다시 광고 요청을 할 경우 새로운 광고가 수신되지 않는다.

- **startAutoRequestAd:**

- (void)startAutoRequestAd:(NSTimeInterval)interval

주어진 interval에 따라 자동으로 광고를 요청한다.

Interval 인자로는 12.0 이상의 값만 넘길 수 있으며, 그보다 작은 값을 사용하는 경우 최소 허용값인 12.0으로 고정되어 동작한다.

광고 효과의 극대화를 위해서는 일반적으로 60초를 사용하는 것을 권장한다.

startAutoRequestAd: 메소드를 호출한 이후에는 stopAutoRequestAd 메소드를 호출해야만 자동요청을 중지할 수 있다.

또한 광고 자동요청을 사용하고 있는 동안에는 requestAd, startAutoRequestAd: 메소드를 호출하여도 새로운 광고가 수신되지 않는다.

#### Parameter

Interval      광고를 자동으로 요청할 시간 간격.

- **stopAutoRequest**

- (void)stopAutoRequestAd

광고 자동요청을 중지한다.

## Properties

---

- **delegate**

@property (nonatomic, assign) id <AdamAdViewDelegate> delegate

AdamAdViewDelegate 프로토콜을 구현한 delegate 객체.

AdamAdViewDelegate 프로토콜의 모든 메소드는 optional이므로, 해당 메소드들을 사용하지 않는다면 할당하지 않아도 무방하다.

일단 delegate 객체를 할당한 이후에는 해당 객체가 메모리에서 해제되지 않도록 주의해야 한다.

delegate 객체가 해제될 때에는 이 속성에 nil 또는 새로운 delegate 객체를 할당해주어야 하며, 그렇지 않은 경우 애플리케이션의 Crash를 유발할 수 있다.

- **clientId**

@property (nonatomic, copy) NSString \*clientId

Daum 으로부터 발급받은 client id 문자열.

필수 속성이며, 정상적인 client id를 할당하지 않는 경우 유효 광고 수신이 불가능하다.

샘플 코드에서는 TestClientId 문자열을 사용하나, 이것은 테스트 용도로만 사용해야 한다.

또한 이 값은 적립금을 쌓는 기준이 되기 때문에, 애플리케이션 배포 전에 발급받은 client id를 정확히 입력했는지 반드시 확인해야 한다.

- **superViewController**

@property (nonatomic, retain) UIViewController \*superViewController

광고 클릭시 광고 페이지를 modalViewController로 화면에 보여줄 부모 뷰 컨트롤러.

이 속성을 따로 설정해주지 않더라도 SDK 내부적으로 적절한 뷰 컨트롤러를 탐색하여 사용하도록 처리되어있다.

따라서 특별한 경우가 아니라면 별도로 뷰 컨트롤러 객체를 할당해주지 않아도 무방하다.

- **transitionStyle**

@property (nonatomic, assign) AdamAdViewTransitionStyle transitionStyle

배너 광고가 새로 수신될 때의 전환 효과.

게임과 같이 그래픽 퍼포먼스가 중요한 애플리케이션의 경우 기본값인 AdamAdViewTransitionStyleNone을 사용하는 것을 권장한다.

- **hasAd**

@property (nonatomic, readonly) BOOL hasAd

현재 광고를 가지고 있는지 여부.

AdamAdView 객체가 현재 광고를 가지고 있는지 여부를 알 수 있는 속성으로, 기본값은 NO이다. 단 한번이라도 성공적으로 광고를 수신하여 현재 화면에 보여지고 있다면 YES 값을 가진다.



- **usingAutoRequest**

@property (nonatomic, readonly) BOOL usingAutoRequest

광고 자동요청을 사용하고 있는지 여부.

AdamAdView 객체가 현재 광고를 자동으로 요청하고 있는지 여부를 알 수 있는 속성으로, 기본값은 NO 이다. startAutoRequest: 메소드를 호출한 이후에는 YES 값을 가지며, stopAutoRequest 메소드를 호출하면 다시 NO 값을 가지게 된다.

- **birth**

@property (nonatomic, copy) NSString \*birth

앱 사용자의 생년월일 정보.

앱 내부에서 사용자의 생년월일 정보를 가지는 경우, birth 속성에 할당하여 타겟팅 광고를 수신 받을 수 있다. 다음과 같은 포맷의 문자열만 할당 가능하며, 유효하지 않은 포맷으로 할당한 경우 무시된다.

- 생년월일 정보를 모두 가진 경우: @"yyyyMMdd"
- 연도 정보만 가진 경우: @"yyyy----"
- 생일 정보만 가진 경우: @"----MMdd"

- **gender**

@property (nonatomic, copy) NSString \*gender

앱 사용자의 성별 정보.

앱 내부에서 사용자의 성별 정보를 가지는 경우, gender 속성에 할당하여 타겟팅 광고를 수신 받을 수 있다. 다음과 같은 포맷의 문자열만 할당 가능하며, 유효하지 않은 포맷으로 할당한 경우 무시된다.

- 남성: @"M"
- 여성: @"F"

- **sdkVersion**

@property (nonatomic, readonly) NSString \*sdkVersion

SDK 버전 정보.

현재 사용 중인 Ad@m SDK의 버전 문자열을 가진다.

## Constants

---

- **AdamAdViewTransitionStyle**

광고 전환효과 스타일.

```
typedef enum {  
    AdamAdViewTransitionStyleNone,  
    AdamAdViewTransitionStyleCurl,  
    AdamAdViewTransitionStyleFlip  
} AdamAdViewTransitionStyle;
```

- ✓ AdamAdViewTransitionStyleNone: 광고 전환효과 없음. 기본값.
- ✓ AdamAdViewTransitionStyleCurl: 광고 영역이 상단으로 말려 올라가며 새 광고가 나타나는 효과.
- ✓ AdamAdViewTransitionStyleFlip: 광고 영역이 반대편으로 뒤집히며 새 광고가 나타나는 효과.

## AdamAdViewDelegate

---

광고 뷰의 동작과 관련하여 발생하는 이벤트가 전달되는 delegate 메소드들이 선언되어있다.

모든 메소드들은 optional이므로, 이 메소드들을 사용하지 않는 경우 AdamAdViewDelegate 프로토콜을 따로 구현해야 할 필요는 없다.

### 광고 수신 / 실패 관련 delegate 메소드

---

- **didReceiveAd:**

- (void)didReceiveAd:(AdamAdView \*)adView

광고 수신 성공시 호출되는 메소드.

#### Parameter

adView      광고 수신 성공 이벤트가 발생한 AdamAdView 객체.

- **didFailToReceiveAd:error:**

- (void)didFailToReceiveAd:(AdamAdView \*)adView error:(NSError \*)error

광고 수신 실패시 호출되는 메소드.

error.domain 값이 AdamErrorDomain이라면, error.code 값을 가지고 AdamErrorType 열거형에 정의된 에러 코드로부터 실패 원인을 알아낼 수 있다.

상세한 에러 메시지를 보고자 한다면 error.localizedDescription 값을 출력해보면 된다.

#### Parameter

adView      광고 수신 실패 이벤트가 발생한 AdamAdView 객체.

error      광고 수신에 실패한 원인이 되는 error 객체.

- **willOpenFullScreenAd:**

- (void)willOpenFullScreenAd:(AdamAdView \*)adView

전체화면 광고가 보여질 때 호출되는 메소드.

배너 광고를 터치하여 광고 페이지가 전체화면에 보여질 때 호출된다.

**Parameter**

adView      광고 페이지 열림 이벤트가 발생한 AdamAdView 객체.

- **didOpenFullScreenAd:**

- (void)didOpenFullScreenAd:(AdamAdView \*)adView

전체화면 광고가 보여진 직후 호출되는 메소드.

배너 광고를 터치하여 광고 페이지가 전체화면에 보여진 직후 호출된다.

**Parameter**

adView      광고 페이지 열림 완료 이벤트가 발생한 AdamAdView 객체.

- **willCloseFullScreenAd:**

- (void)willCloseFullScreenAd:(AdamAdView \*)adView

전체화면 광고가 닫힐 때 호출되는 메소드.

전체화면으로 보여지고 있는 광고 페이지가 닫힐 때 호출된다.

**Parameter**

adView      광고 페이지 닫힘 이벤트가 발생한 AdamAdView 객체.

- **didCloseFullScreenAd:**

- (void)didCloseFullScreenAd:(AdamAdView \*)adView

전체화면 광고가 닫힌 직후 호출되는 메소드.

전체화면으로 보여지고 있는 광고 페이지가 닫히고 난 직후 호출된다.

**Parameter**

adView      광고 페이지 닫힘 완료 이벤트가 발생한 AdamAdView 객체.

- **willResignByAd:**

- (void)willResignByAd:(AdamAdView \*)adView

광고 터치로 인해 애플리케이션이 종료될 때 호출되는 메소드.

배너 광고를 터치하여 전화 걸기 또는 앱스토어로 이동하는 경우, 애플리케이션이 백그라운드로 들어가게 될 때 호출된다.

**Parameter**

adView      백그라운드로 전환 이벤트를 발생시킨 AdamAdView 객체.

### 객체 생성 메소드

---

- **sharedInterstitial**

+ (AdamInterstitial \*)sharedInterstitial

AdamInterstitial 클래스의 Singleton 객체인 sharedInterstitial을 리턴한다.

**Return**

AdamInterstitial 객체.

### 광고 요청 메소드

---

- **requestAndPresent**

- (void)requestAndPresent

전면형 광고를 요청하고, 요청에 대한 수신이 성공적으로 이루어지면 즉시 화면에 노출한다.

최소 호출 간격은 3분이며, 이전 요청의 성공/실패 여부와 상관없이 3분 이내에는 다시 호출할 수 없다.

## Properties

---

- **delegate**

@property (nonatomic, assign) id <AdamInterstitialDelegate> delegate

AdamInterstitialDelegate 프로토콜을 구현한 delegate 객체.

AdamInterstitialDelegate 프로토콜의 모든 메소드는 optional이므로, 해당 메소드들을 사용하지 않는다면 할당하지 않아도 무방하다.

일단 delegate 객체를 할당한 이후에는 해당 객체가 메모리에서 해제되지 않도록 주의해야 한다.

delegate 객체가 해제될 때에는 이 속성에 nil 또는 새로운 delegate 객체를 할당해주어야 하며, 그렇지 않은 경우 애플리케이션의 Crash를 유발할 수 있다.

- **clientId**

@property (nonatomic, retain) NSString \*clientId

Daum 으로부터 발급받은 client id 문자열.

필수 속성이며, 정상적인 client id를 할당하지 않는 경우 유효 광고 수신이 불가능하다.

샘플 코드에서는 InterstitialTestClientId 문자열을 사용하나, 이것은 테스트 용도로만 사용해야 한다.

또한 이 값은 적립금을 쌓는 기준이 되기 때문에, 애플리케이션 배포 전에 발급받은 client id를 정확히 입력했는지 반드시 확인해야 한다.

- **superViewController**

@property (nonatomic, retain) UIViewController \*superViewController

광고 수신 성공시 광고 페이지를 modalViewController로 화면에 보여줄 부모 뷰 컨트롤러.

이 속성을 따로 설정해주지 않더라도 SDK 내부적으로 적절한 뷰 컨트롤러를 탐색하여 사용하도록 처리되어있다. 따라서 특별한 경우가 아니라면 별도로 뷰 컨트롤러 객체를 할당해주지 않아도 무방하다.

- **birth**

@property (nonatomic, copy) NSString \*birth

앱 사용자의 생년월일 정보.

앱 내부에서 사용자의 생년월일 정보를 가지는 경우, birth 속성에 할당하여 타겟팅 광고를 수신 받을 수 있다. 다음과 같은 포맷의 문자열만 할당 가능하며, 유효하지 않은 포맷으로 할당한 경우 무시된다.

- 생년월일 정보를 모두 가진 경우: @"yyyyMMdd"
- 연도 정보만 가진 경우: @"yyyy----"
- 생일 정보만 가진 경우: @"----MMdd"

- **gender**

@property (nonatomic, copy) NSString \*gender

앱 사용자의 성별 정보.

앱 내부에서 사용자의 성별 정보를 가지는 경우, gender 속성에 할당하여 타겟팅 광고를 수신 받을 수 있다. 다음과 같은 포맷의 문자열만 할당 가능하며, 유효하지 않은 포맷으로 할당한 경우 무시된다.

- 남성: @"M"
- 여성: @"F"

- **sdkVersion**

@property (nonatomic, readonly) NSString \*sdkVersion

SDK 버전 정보.

현재 사용 중인 Ad@m SDK의 버전 문자열을 가진다.



## AdamInterstitialDelegate

---

AdamInterstitial 객체와 관련된 이벤트가 전달되는 delegate 메소드들이 선언되어있다.

모든 메소드들은 optional이므로, 이 메소드들을 사용하지 않는 경우 AdamInterstitialDelegate 프로토콜을 따로 구현해야 할 필요는 없다.

### 광고 수신 / 실패 관련 delegate 메소드

---

- **didReceiveInterstitialAd:**

- (void)didReceiveInterstitialAd:(AdamInterstitial \*)interstitial

전면형 광고 수신 성공시 호출되는 메소드.

#### Parameter

interstitial	광고 수신 성공 이벤트가 발생한 AdamInterstitial 객체.
--------------	--

- **didFailToReceiveInterstitialAd:error:**

- (void)didFailToReceiveInterstitialAd:

(AdamInterstitial \*)interstitial error:(NSError \*)error

전면형 광고 수신 실패시 호출되는 메소드.

광고 수신에 실패한 원인을 알고자 하는 경우, error.localizedDescription 값을 출력해보면 된다.

#### Parameter

interstitial	광고 수신 실패 이벤트가 발생한 AdamInterstitial 객체.
--------------	--

error	광고 수신에 실패한 원인이 되는 error 객체.
-------	-----------------------------

### 광고 노출 관련 delegate 메소드

---

- **willOpenInterstitialAd:**

- (void)willOpenInterstitialAd:(AdamInterstitial \*)interstitial

전면형 광고가 보여질 때 호출되는 메소드.

**Parameter**

interstitial      전면형 광고 열림 이벤트가 발생한 AdamInterstitial 객체.

- **didOpenInterstitialAd:**

- (void)didOpenInterstitialAd:(AdamInterstitial \*)interstitial

전면형 광고가 보여진 직후 호출되는 메소드.

**Parameter**

interstitial      전면형 광고 열림 완료 이벤트가 발생한 AdamInterstitial 객체.

- **willCloseInterstitialAd:**

- (void)willCloseInterstitialAd:(AdamInterstitial \*)interstitial

전면형 광고가 닫힐 때 호출되는 메소드.

**Parameter**

interstitial      전면형 광고 닫힘 이벤트가 발생한 AdamInterstitial 객체.

- **didCloseInterstitialAd:**

- (void)didCloseInterstitialAd:(AdamInterstitial \*)interstitial

전면형 광고가 닫힌 직후 호출되는 메소드.

**Parameter**

interstitial      전면형 광고 닫힘 완료 이벤트가 발생한 AdamInterstitial 객체.

- **willResignByInterstitialAd:**

- (void)willResignByInterstitialAd:(AdamInterstitial \*)interstitial

전면형 광고 내부의 인터랙션에 의해 애플리케이션이 종료될 때 호출되는 메소드.

전면형 광고 내부에서 전화 걸기 또는 앱스토어로 이동 기능이 실행되는 경우, 애플리케이션이 백그라운드로 들어가게 될 때 호출된다.

#### Parameter

interstitial      전면형 광고 닫힘 완료 이벤트가 발생한 AdamInterstitial 객체.

### Constants

---

- **AdamErrorType**

광고 수신 실패에 대한 에러코드.

```
typedef enum {  
    AdamErrorTypeUnknown,  
    AdamErrorTypeNoFillAd,  
    AdamErrorTypeNoClientId,  
    AdamErrorTypeTooSmallAdView,  
    AdamErrorTypeInvisibleAdView,  
    AdamErrorTypeAlreadyUsingAutoRequest,  
    AdamErrorTypeTooShortRequestInterval,  
    AdamErrorTypePreviousRequestNotFinished,  
    AdamErrorTypeOpenedAdExists  
} AdamErrorType;
```

- ✓ AdamErrorTypeUnknown: 원인을 알 수 없는 에러.
- ✓ AdamErrorTypeNoFillAd: 현재 노출 가능한 광고가 없음.
- ✓ AdamErrorTypeNoClientId: ClientId가 설정되지 않았음.
- ✓ AdamErrorTypeTooSmallAdView: 광고 뷰의 크기가 기준보다 작음.
- ✓ AdamErrorTypeInvisibleAdView: 광고 뷰가 화면에 보여지지 않고 있음.
- ✓ AdamErrorTypeAlreadyUsingAutoRequest: 이미 광고 자동요청 기능을 사용하고 있는 상태임.
- ✓ AdamErrorTypeTooShortRequestInterval: 허용되는 최소 호출 간격보다 짧은 시간 내에 광고를 재요청 했음.
- ✓ AdamErrorTypePreviousRequestNotFinished: 이전 광고 요청에 대한 처리가 아직 완료되지 않았음.
- ✓ AdamErrorTypeOpenedAdExists: 배너가 확장되어 열려있는 상태임.

### 광고 수신 실패 원인에 따른 처리 방법

AdamAdView 객체에서 광고 수신 실패시에는 AdamAdViewDelegate 프로토콜의 `didFailToReceiveAd:error:` 메소드가 호출되고, AdamInterstitial 객체에서 광고 수신 실패시에는 AdamInterstitialDelegate 프로토콜의 `didFailToReceiveInterstitialAd:error:` 메소드가 호출된다. 이 메소드로부터 전달받은 error 객체를 가지고 광고 수신 실패의 원인을 파악할 수 있으며, 각각의 에러 타입에 대한 처리는 다음과 같이 한다.

#### 1. No Fill Ad (AdamAdView / AdamInterstitial 공통)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypeNoFillAd
- ✓ **원인:**  
광고를 정상적으로 요청하였으나, 광고 서버에서 보내줄 수 있는 유효 광고가 없을 경우 발생한다. 또는 유효하지 않은 Client ID를 설정한 경우에도 발생한다.
- ✓ **처리 방법:**  
배너 광고의 경우 <http://mobile.biz.daum.net> 에서 '사이트/앱 등록' 메뉴에 접속하면 '하우스애드'를 등록할 수 있다. 하우스애드란 자신의 앱에 자체 광고를 노출할 수 있는 기능으로, 광고 서버에서 유효 광고를 보내줄 수 없는 경우 자신이 등록한 하우스애드가 수신된다.  
Interstitial 광고의 경우에는 하우스 애드가 지원되지 않으므로, 일정 시간 이후 다시 호출해본다.

#### 2. No Client Id (AdamAdView / AdamInterstitial 공통)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypeNoClientId
- ✓ **원인:**  
AdamAdView 또는 AdamInterstitial 객체의 `clientId` 속성을 지정하지 않고 광고를 호출한 경우 발생한다.
- ✓ **처리 방법:**  
AdamAdView 또는 AdamInterstitial 객체의 `clientId` 속성에 발급받은 Client ID 문자열을 정확하게 할당한다.

### 3. Too Small Ad View (AdamAdView)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypeTooSmallAdView
- ✓ **원인:**  
광고 뷰가 기준 사이즈보다 작은 상태에서 광고를 호출한 경우 발생한다.
- ✓ **처리 방법:**  
광고 뷰의 기준 사이즈는 320px \* 48px 이다. 광고 뷰의 width는 320px 이상, height는 48px 이상으로 지정한다.

### 4. Invisible Ad View (AdamAdView)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypeInvisibleAdView
- ✓ **원인:**  
광고 뷰가 화면에 보이지 않는 상태에서 광고를 호출한 경우 발생한다. 부모 뷰에 붙어있지 않은 경우, modalViewController에 의해 가려진 경우 등이 해당된다.
- ✓ **처리 방법:**  
광고 뷰가 화면에 정상적으로 디스플레이 되고 있는지 확인한다.

### 5. Already Using Auto Request (AdamAdView)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypeAlreadyUsingAutoRequest
- ✓ **원인:**  
광고 뷰가 Auto Request 기능을 사용하고 있는 상태에서 별도로 광고 요청을 한 경우 (requestAd, startAutoRequestAd: 메소드 호출) 발생한다.
- ✓ **처리 방법:**  
startAutoRequestAd: 메소드를 호출하여 광고를 자동으로 요청하고 있을 때에는 해당 기능에 전적으로 광고 요청을 맡기고, 별도로 광고를 요청하지 않는다.

## 6. Too Short Request Interval (AdamAdView / AdamInterstitial 공통)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypeTooShortRequestInterval
- ✓ **원인:**  
AdamAdView 객체는 광고 요청 후 12초 이내에 광고를 재요청 하는 경우,  
AdamInterstitial 객체는 광고 요청 후 3분 이내에 광고를 재요청 하는 경우 발생한다.
- ✓ **처리방법:**  
AdamAdView에서 타이머를 이용해 requestAd 메소드를 직접 호출하는 경우, 타이머의 interval을 적절한 값으로 조정한다. 또는 startAutoRequest: 메소드를 사용하여 광고를 자동 요청하도록 한다.  
AdamInterstitial 객체의 경우에는 3분 이내에 광고를 재요청 하지 않도록 호출 시점을 조정한다.

## 7. Previous Request Not Finished (AdamAdView / AdamInterstitial 공통)

- ✓ **domain:** AdamErrorDomain
- ✓ **code:** AdamErrorTypePreviousRequestNotFinished
- ✓ **원인:**  
이전 광고 요청에 대한 처리가 완료되지 않은 상태에서 광고를 재요청 하는 경우 발생한다. 네트워크 상태가 매우 좋지 않은 경우 발생할 수 있다.
- ✓ **처리방법:**  
이 에러는 네트워크 상태가 좋지 않은 경우 일시적으로 발생 가능한 에러이므로, 별도로 처리를 해줄 필요는 없다. 해당 에러가 자주 발생한다면 광고 요청 interval을 다소 길게 조정하도록 한다.