



iOS SDK User Guide 3.8

SMART
mediation

SMART
ad

SMART
dialog

 Banner

리워드 배너

 Link

리워드 링크

개요 mediation

실제 프로젝트환경에 SDK 적용을 위한 문서입니다.

SDK 다운부터 API 키 발급까지에 대한 과정은 아래 링크에서 확인하실 수 있습니다.

<http://adlibr.com/howto/prepare>

애드립을 통해 사용할 플랫폼은 프로젝트에서 선택적으로 포함하여 최종 바이너리 크기를 줄일 수 있습니다.

실제 테스트 프로젝트를 컴파일 하기 위하여

각 플랫폼 사이트에서 발급받은 APP - ID 및 각 OS 에 맞는 최신 SDK가 별도로 필요합니다.

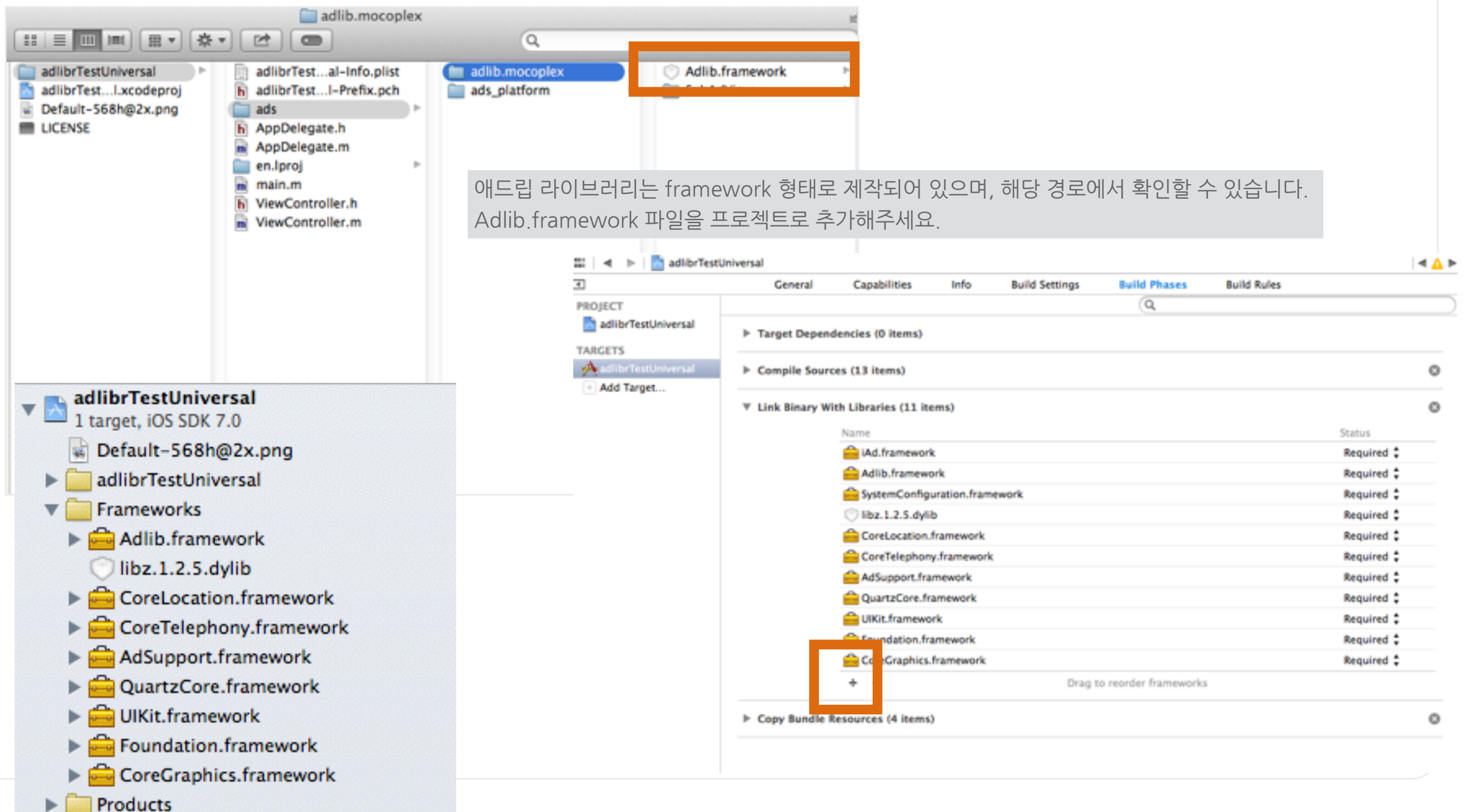
기본적으로 테스트 프로젝트는 각 플랫폼의 jar 파일만 새로 링크하면 동작하도록 제작되었으며
실제 플랫폼의 SDK 작동방식이 변하지 않는 한 SubAdView 안의 내용을 수정할 필요는 없습니다.
(실제 발급받은 광고 플랫폼 ID 적용 부분은 수정해야 합니다.)

권장 개발환경

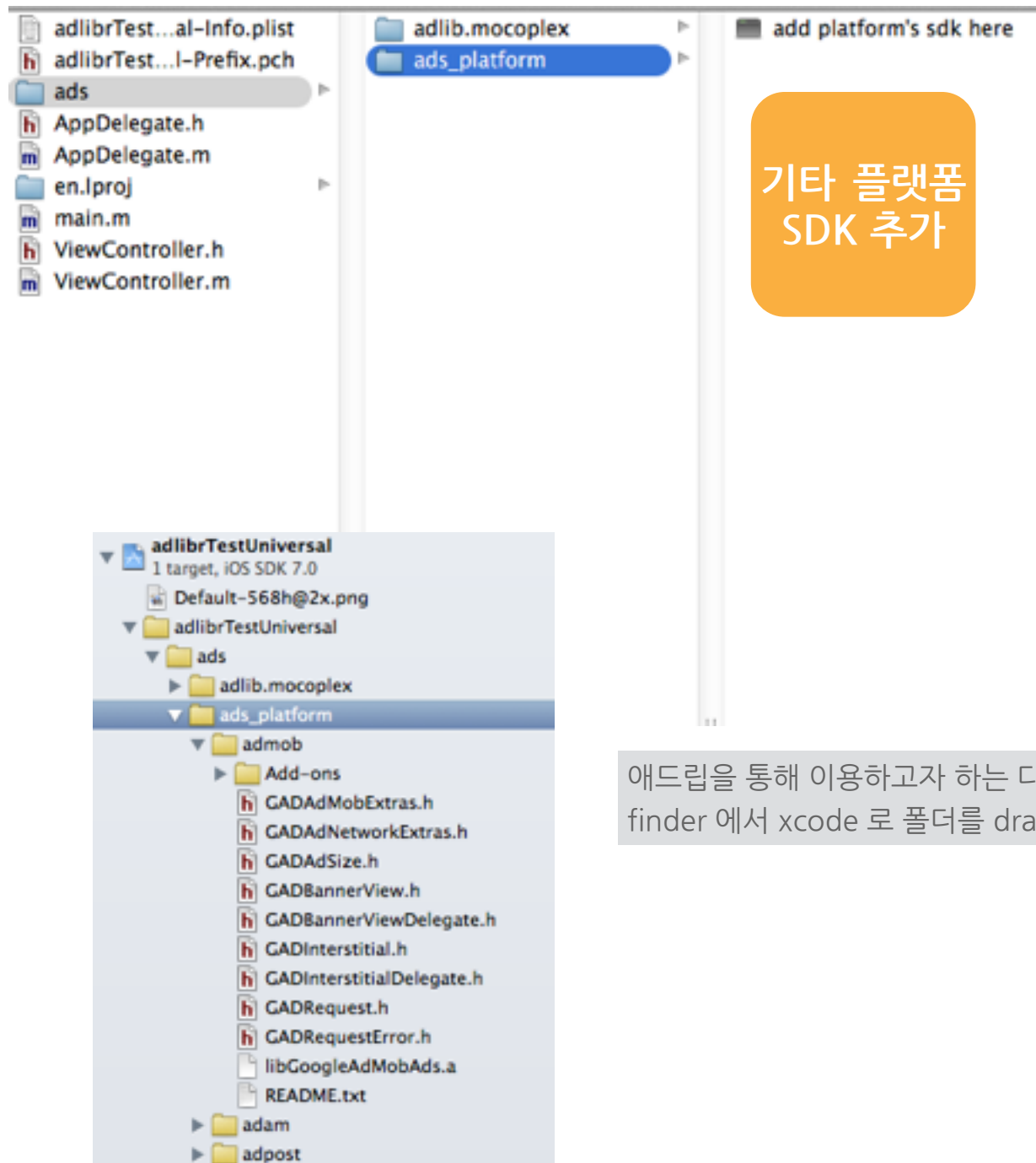
- Xcode 5.0 이상
- Base SDK 7.0 이상
- iOS Deployment Target 4.3 이상

컴파일을 위한 준비 #1

각 플랫폼의 가입과 SDK 다운로드를 개별적으로 진행합니다.
SDK 다운부터 API 키 발급까지에 대한 과정은 아래 링크에서 확인하실 수 있습니다.
<http://adlibr.com/howto/prepare>

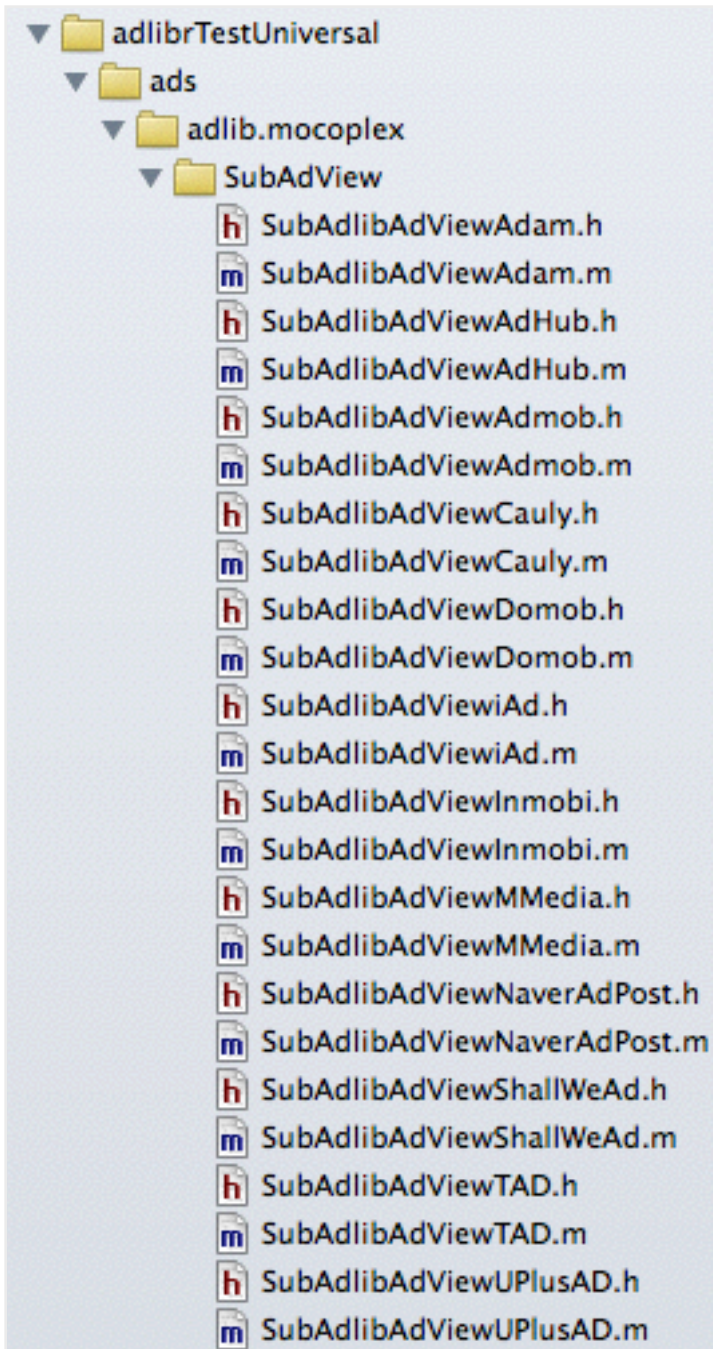


컴파일을 위한 준비 #2



애드립을 통해 이용하고자 하는 다른 플랫폼의 라이브러리를 아래 경로로 모두 추가합니다.
finder 에서 xcode 로 폴더를 drag & drop 하면 손쉽게 모든 파일을 추가할 수 있습니다.

프로젝트 자세히



프로젝트의 SubAdView 폴더안에
각 광고 플랫폼의 실제 구현부 소스가 위치해 있습니다.

라이브러리 추가와 마찬가지로
실제 사용할 클래스만 선택, 포함하여 최종 파일 크기를 줄일 수 있습니다.

실제 프로젝트 연동 #1

AppDelegate.m

애드립을 연동하기 위해서 초기 구현부를 아래와 같이 수정합니다.

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self initAdlib];

    .....
}
```

- (void)initAdlib

```
NSString* adlibKey = @"ADLIB-API-KEY";
```

```
[[AdlibManager sharedInstance] initAdlib:adlibKey];
```

```
// 쓰지 않을 광고 플랫폼은 삭제해주세요.
```

```
[[AdlibManager sharedInstance] setPlatform:@"ADAM" withClass:[SubAdlibAdViewAdam class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"ADMOB" withClass:[SubAdlibAdViewAdmob class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"CAULY" withClass:[SubAdlibAdViewCauly class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"TAD" withClass:[SubAdlibAdViewTAD class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"IAD" withClass:[SubAdlibAdViewiAd class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"NAVER" withClass:[SubAdlibAdViewNaverAdPost class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"SHALLWEAD" withClass:[SubAdlibAdViewShallWeAd class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"INMOBI" withClass:[SubAdlibAdViewInmobi class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"MMEDIA" withClass:[SubAdlibAdViewMMedia class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"DOMOB" withClass:[SubAdlibAdViewDomob class]];
```

```
[[AdlibManager sharedInstance] setPlatform:@"UPLUSAD" withClass:[SubAdlibAdViewUPlusAD class]];
```

```
// 쓰지 않을 광고 플랫폼은 삭제해주세요.
```

1. 먼저 애드립에서 발급받은 API - KEY 를 연결합니다.
2. 실제 광고 구현 Class 를 애드립에 위와같이 연결합니다.

실제 프로젝트 연동 #1 자세히

1. 애드립 에서 발급받은 API KEY 를 설정합니다. (필수)

2. 실제 구현된 광고 뷰는 `SubAdLibAdViewCore` 를 상속받아 구현됩니다. (필수)

SubAdView 폴더 안에 각 광고뷰에 대한 구현 소스가 들어 있으며, 필요에 의해 추가적인 구현으로 더 다양한 기능 구현이 가능합니다.

실제 프로젝트 연동 #2 - 1

애드립을 연동하기 위해 ViewController에서는 크게 3가지 구현부를 가집니다.

<https://github.com/mocoplex/adlibr-SDK-ios/blob/master/adlibrTestUniversal/adlibrTestUniversal/ViewController.m>

1. 뷰가 화면에 보이는 시점에 애드립에 현재 컨트롤러와 광고가 올라갈 뷰를 연결합니다.

```
-(void)viewDidAppear:(BOOL)animated
{
    [[AdlibManager sharedInstance] attach:self
                                withView:self.view
                                withReceiver:@selector(layoutForAD)
                                useHouseBanner:YES];
}
```

useHouseBanner : 초기 광고 로딩 전에 기본 하우스배너를 보여줄지 말지 여부. (생략가능)
YES 이면 애드립 배너 혹은 대쉬보드에서 설정하신 사용자배너가 초기 로딩 전에 보여집니다. (기본값은 YES)

2. 뷰가 화면에서 사라지는 시점에 애드립 컨트롤러 연결을 끊습니다.

```
-(void)viewWillDisappear:(BOOL)animated
{
    [[AdlibManager sharedInstance] detach:self];
}
```


실제 프로젝트 연동 #2 - 1 계속

3. 광고를 수신했을때, 크기가 변경되었을때, 회전이 일어났을때 추가적인 구현을 위해 layout 을 관리할 구현부를 추가합니다.

```
// 광고를 수신했다. 광고view 의 크기와 위치를 재설정한다.  
- (void)layoutForAD  
{  
    // 광고 view의 크기  
    CGSize sz = [[AdlibManager sharedInstance] size];  
  
    .....  
  
    [[AdlibManager sharedInstance] moveAdContainer:adPoint];  
}
```

layout 재설정에 대한 내용은 실제 테스트 프로젝트의 실행을 통해 좀 더 자세히 확인할 수 있습니다.

기본적으로 애드립은 모든 광고뷰에 대한 회전을 지원하며, 이와 관련한 별도 추가 구현이 필요하지 않습니다.
회전과 같은 UI 변경이 일어났을 때 UI 를 재정비하기 위한 액션을 위 layoutForAD 에서 수행합니다.

실제 프로젝트 연동 #2 - 2

```
-(void)viewDidAppear:(BOOL)animated
{
    // 광고뷰를 연결합니다. 페이지 이름을 설정하면 세부적인 설정으로 페이지마다 노출 정책을 다르게 설정할 수 있습니다.
    // 페이지별 설정 메뉴에서 상세 정책을 지정, 적용할 수 있습니다.
    [[AdlibManager sharedInstance] attach:self withView:self.view
                                     withReceiver:@selector(whenGotAd) withPageName:@"xxxx"
                                     useHouseBanner:YES];
}
```

useHouseBanner : 초기 광고 로딩 전에 기본 하우스배너를 보여줄지 말지 여부. (생략가능)
YES 이면 애드립 배너 혹은 대쉬보드에서 설정하신 사용자배너가 초기 로딩 전에 보여집니다. (기본값은 YES)

실제 페이지에 이름을 설정하여 좀더 세부적인 정책을 지정, 적용할 수 있습니다.
페이지별 설정을 하기 위해서는 위와같이 페이지 이름이 미리 지정되어 있어야 합니다.

 Mediation
전역설정

 Mediation
페이지 설정

 Mediation
설정히스토리 보기

 Dialog
대화상자

 Custom AD
사용자 광고 설정

 etc
클라이언트
버전 설정

- 페이지별로 노출할 광고 플랫폼을 지정할 수 있습니다.
- 설정되지 않은 페이지는 전역설정에서의 설정대로 적용됩니다.
- 실제 프로그램 구현부에서 페이지 이름을 지정하여야 합니다(매뉴얼을 참조해주세요).

Page	설정내용
page1	Adam > Admob > iAd > Naver
page2	Inmobi > T-ad > ShallWeAd > Cauly > 사용자 배너
xxxx	Adam > Admob > iAd > Naver > 사용자 배너

페이지 추가하기

실제 프로젝트 연동 #3

SubAdView 안의 구현부는 실제 광고 플랫폼의 뷰를 포함하는 Class 이며 여기서 각 광고 플랫폼의 아이디를 설정해야 합니다.

```
#import "SubAdlibAdViewInmobi.h"

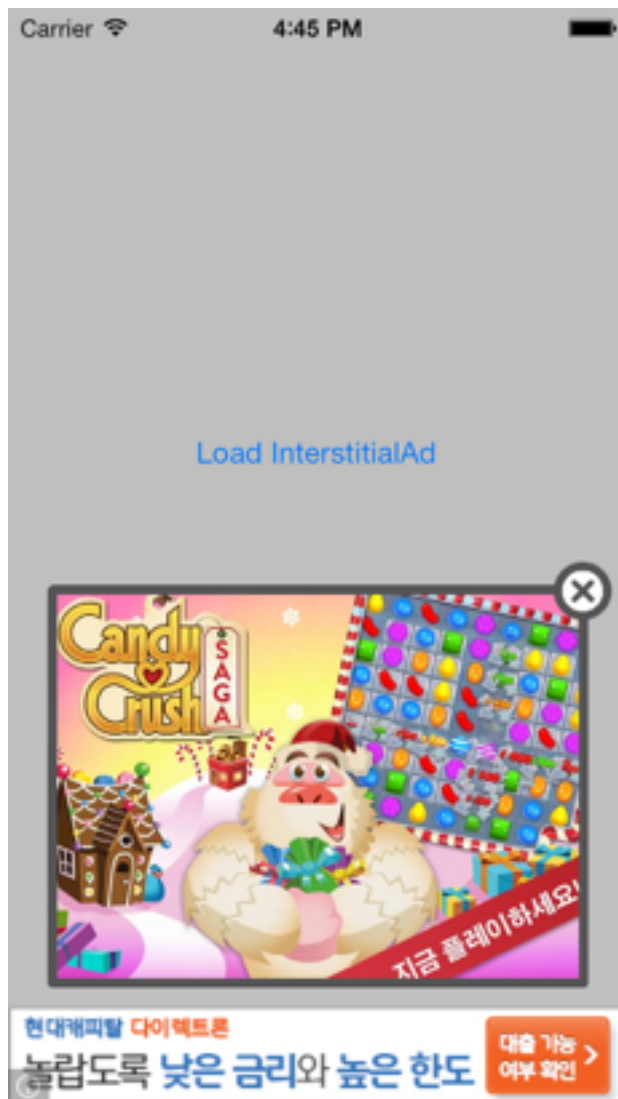
// INMOBI의 APP 아이디를 설정합니다.
#define INMOBI_ID @"INMOBI APP ID"

@implementation SubAdlibAdViewInmobi
```

각 플랫폼에서 자신이 등록한 앱에 대해 발급받은 아이디를 위 영역에 입력합니다.

기타 플랫폼도 마찬가지로 방법으로 ID를 추가합니다.

애드립 팝 배너



- 그림과 같은 형태의 배너가 노출된 후 일정 시간이 지나면 사라집니다.

- 아래의 메소드를 호출하여 애드립 팝 배너를 보여줄 수 있습니다.

```
//(optional) 프레임 컬러 설정 (UIColor*)
[[AdlibManager sharedInstance] setAdlibPopFrameColor:[UIColor darkGrayColor]];
//(optional) 버튼 컬러 설정 (AdlibPopBtnStyle)
[[AdlibManager sharedInstance] setAdlibPopCloseButtonStyle:AdlibPopBtnStyleWhite];
//(optional) in, out 애니메이션 설정 (AdlibPopAnimationType)
[[AdlibManager sharedInstance] setAdlibPopInAnimation:AdlibPopAnimationTypeSlide
                                outAnimation:AdlibPopAnimationTypeNone];
//애드립 팝배너 보이기 (align: AdlibPopAlign) - align값 기준으로 padding만큼 떨어진 위치에 보여줍니다.
[[AdlibManager sharedInstance] showAdlibPopBanner:AdlibPopAlignBottom
                                withPadding:60
                                withDelegate:self];
```

- 아래의 메소드를 호출하여 애드립 팝 배너를 숨길 수 있습니다.

```
[[AdlibManager sharedInstance] hideAdlibPopBanner];
```

애드립 팝 배너

팝배너 관련 delegate 메소드는 아래와 같습니다. (optional)

```
-(void)didReceiveAdlibPopAd
{
    NSLog(@"%@", @"Pop banner 수신 성공!!");
}

-(void)didFailToReceiveAdlibPopAd
{
    NSLog(@"%@", @"Pop banner 수신 실패..");
}

-(void)didCloseAdlibPopAd
{
    NSLog(@"%@", @"Pop banner 닫힘..");
}
```

애드립 팝 배너

팝배너 설정 관련

팝배너 버튼 내부 색상 (기본값: AdlibPopBtnStyleWhite)

```
typedef enum {  
    AdlibPopBtnStyleWhite,  
    AdlibPopBtnStyleBlack  
} AdlibPopBtnStyle;
```

팝배너 위치 기준 (기본값: AdlibPopAlignBottom)

```
typedef enum {  
    AdlibPopAlignLeft,  
    AdlibPopAlignTop,  
    AdlibPopAlignRight,  
    AdlibPopAlignBottom  
} AdlibPopAlign;
```

팝배너 애니메이션 타입 (기본값: AdlibPopAnimationTypeSlide)

```
typedef enum {  
    AdlibPopAnimationTypeNone,  
    AdlibPopAnimationTypeSlide  
} AdlibPopAnimationType;
```

애드립 리워드 배너 (띠배너)

Banner

리워드 배너

- Background request 를 통해 지면낭비 없이, 더 높은 단가의 광고를 우선적으로 노출합니다.
- 별도 플랫폼의 회원가입이나 SDK 연동 없이, 바로 제휴 플랫폼의 광고를 효율적으로 노출할 수 있습니다.
- 애드립 리워드 배너는 실시간 단가 정보를 활용하여, 기존보다 높은 단가의 광고를 우선 노출하기 때문에 기존보다 더 높은 광고수익을 기대할 수 있습니다.

AdlibTest iOS



설정 후 저장 및 적용하기 버튼을 눌러야 반영됩니다.

저장 및 적용하기

- 대쉬보드의 전역설정에서 R Banner 단가를 Auto 혹은 원하는 단가로 설정 후 “ON” 선택하여 “저장 및 적용하기” 버튼을 클릭합니다.
- 위 설정을 키는 것 만으로 바로 리워드배너를 앱에 노출할 수 있습니다.
- 리워드 배너 노출을 위해 코드상에 추가로 구현할 내용은 없습니다.

애드립 리워드 배너(전면배너)

Banner

리워드 배너



- 대쉬보드의 간단한 설정으로 “앱 시작시”에 전면배너를 자동 노출 할 수 있습니다.
- 그 외의 시점이나, 버튼 클릭 등과 같은 이벤트 발생 시에는 직접 호출하여 전면배너를 노출 할 수 있습니다.
- 직접 호출하여 전면배너를 노출할 때는 다양한 플랫폼을 스케줄링 할 수 있습니다.

애드립 리워드 배너(전면배너)

Banner

리워드 배너

1. 자동노출

전면배너 간편 노출 설정

- 리워드배너 슬롯을 통해, 리포트 및 노출 스케줄설정 등, 다양한 기능이 지원되는 "사용자 리워드 전면배너" 기능을 이용할 수 있습니다.
- 별도의 코드상의 추가 구현없이 설정한 시점에 자동으로 전면배너가 노출됩니다. (SDK 3.5 이상)
- 트래픽 및 수익 내역은 D+1 "리워드 배너" 수익 리포트에서 확인할 수 있습니다.
- 설정한 시점외에, `[[AdlibManager] loadInterstitialAd: withDelegate:]` 를 호출하면 해당 시점에 광고가 노출됩니다. (매뉴얼 참고)



앱 시작시

ON

전면배너 스케줄링을 사용하려면, 여기를 클릭해주세요.

테스트 전면배너를 노출하려면, 여기를 클릭해주세요.

- 대쉬보드에서 "앱 시작시" 를 ON으로 설정하시면, 아무런 구현 없이 해당 시점에 전면배너가 노출 됩니다.
- 단, 자동 노출 기능은 스케줄링을 사용할 수 없으며, 애드립에서 제공하는 전면배너 노출만 가능합니다.

애드립 리워드 배너(전면배너)

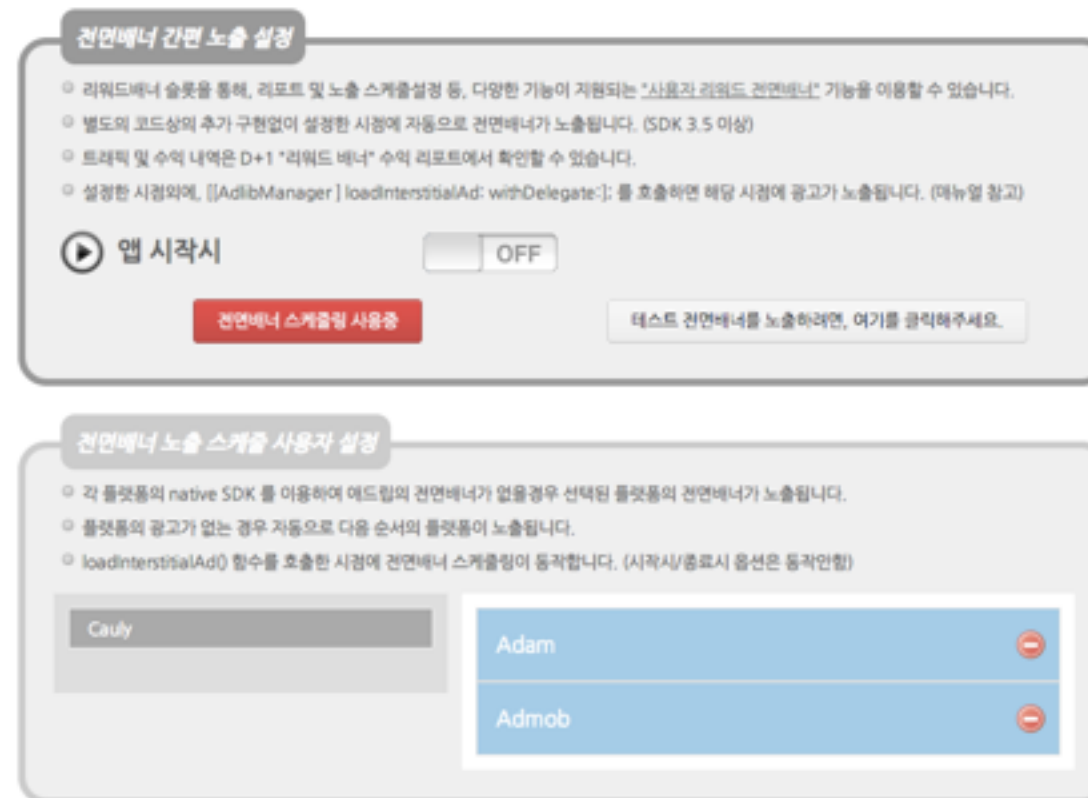
Banner

리워드 배너

2. 직접 호출

전면배너 자동노출을 사용하지 않고 직접 호출을 하기 위해서는 아래 코드를 입력해주세요.

```
[[AdlibManager sharedInstance] loadInterstitialAd:self withDelegate:self];
```



- 대쉬보드에서 전면배너 스케줄링 사용을 켜면, 스케줄 설정을 할 수 있습니다.
- 전면광고 수신에 실패하면 애드립전면배너 + 설정된 플랫폼 순서대로 전면광고를 호출합니다.

애드립 리워드 배너(전면배너)

 Banner

리워드 배너

전면배너 관련 delegate 메소드는 아래와 같습니다. (from : 광고 플랫폼 이름)

```
-(void)didReceiveAdlibInterstitialAd:(NSString*)from
{
    NSLog(@"%@ Interstitial Ad 수신 성공!!", from);
}

-(void)didFailToReceiveAdlibInterstitialAd:(NSString*)from
{
    NSLog(@"%@ Interstitial Ad 수신 실패..", from);
}

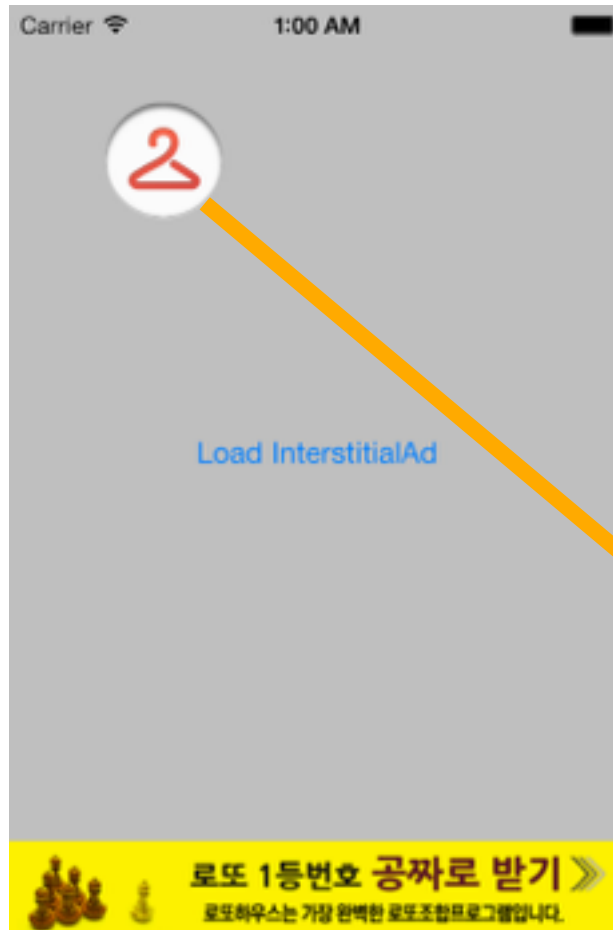
-(void)didCloseAdlibInterstitialAd:(NSString*)from
{
    NSLog(@"%@ Interstitial Ad 닫힘..", from);
}

-(void)didFailToReceiveAllInterstitialAd:(NSString*)from
{
    NSLog(@"%@, @"스케줄링된 모든 Interstitial Ad 수신 실패");
}
```

애드립 리워드 링크



애드립 리워드링크 개요



- 기존 Mediation 시스템과 별도로 동작하는 시스템입니다.
- 광고의 자유로운 위치 설정이 가능하며, 아이콘 형태로 최소한의 지면을 사용합니다.
- 배너광고 이외의 수익을 추가로 더 얻을 수 있습니다.
- 노출 여부를 관리자 페이지에서 손쉽게 설정할 수 있습니다.



애드립 리워드 링크



애드립 리워드 링크는 기존의 mediation 시스템과 개별적으로 동작합니다.

애드립 리워드 링크를 이용해, 앱에 [아이콘] 형태의 광고를 노출할 수 있으며,
[아이콘] 광고를 이용하여 배너 수익외 추가 수익을 거둘 수 있습니다.

리워드 링크의 카테고리 상품은 계속적으로 확대 적용될 예정이며, 현재 [리워드 게임 링크]를 제공하고 있습니다.

애드립 리워드 링크를 앱에 적용하기 위해서는



메뉴를 통해서 리워드 링크를 추가하여 API 키를 발급받습니다.

리워드링크의 정보를 수정해주세요.

링크 이름	<input type="text"/>
앱 (마켓주소) URL	<input type="text" value="http://"/>
링크 ID	<input type="text"/>
노출 카테고리	<input type="text" value="GAME"/>
아이콘 Width (dp)	<input type="text" value="60"/>
아이콘 Height (dp)	<input type="text" value="60"/>
노출 여부	<input checked="" type="checkbox"/>

설정을 통해 아이콘 사이즈와 노출여부를 손쉽게 변경할 수 있습니다.

리워드링크 사용법 - 구현

리워드링크 아이콘을 노출하기 원하는 화면에 아래 두가지 상황에 맞추어 method 를 호출하여 아이콘을 attach / detach 합니다.

```
-(void)viewDidAppear:(BOOL)animated
{
    // 리워드링크 적용 예
    // addRewardLinkIcon을 추가하면 아이콘을 여러개 붙일 수 있습니다. 단 한 view에서 같은 키의 아이콘을 사용할 수 없습니다.
    [[AdlibRewardLink sharedInstance] attachRewardLink:self withView:self.view];
    [[AdlibRewardLink sharedInstance] addRewardLinkIcon:@"REWARD_LINK_ID" atPoint:CGPointMake(x, y)];
}

-(void)viewWillDisappear:(BOOL)animated
{
    // 리워드링크
    [[AdlibRewardLink sharedInstance] detachRewardLink:self];
}
```

Smart Dialog 란?



- 공지사항 알림, 앱 업데이트 유도 등 개인화된 대화상자를 별도 구현없이 손쉽게 생성 할 수 있습니다.
- 온라인에서의 간단한 설정만으로 손쉽게 대화상자를 컨트롤 할 수 있습니다.
- 대화상자를 통해 배너가 노출될 때 마다 3,000여개의 앱이 사용중인 애드립 네트워크 안에서 나의 앱도 교차 홍보되어, 더 많은 사용자들에게 나의 앱을 알릴 수 있습니다.



- 대화상자는 대쉬보드의 대화상자 탭에서 “대화상자 추가하기” 를 클릭 후 원하는 형태로 설정하여 노출할 수 있습니다.
- 노출시점 설정시, App 시작시점은 프로젝트에서의 소스 구현분의 `[[AdlibManager sharedInstance] initAdlib:adlibKey];` 해당 소스가 수행된 직후입니다.

자주 묻는 질문 FAQ 와 팁 TIP

- 각 나라에 맞게 광고 스케줄을 달리 적용하고 싶습니다. (지역 타게팅)

애드립의 앱 스케줄 설정 화면에서, 타게팅을 원하는 나라를 추가하여 손쉽게 노출전략을 설정할 수 있습니다.

정리

애드립 연동이 마무리되었습니다.

실제 프로젝트에 애드립을 연동하기 위해

1. 각 플랫폼의 ID 발급 및 SDK 다운로드
2. SDK 프로젝트에 라이브러리 추가
3. `-(void)initAdlib` 수정 (애드립 API KEY 설정)
4. SubAdView 안의 광고 ID 설정 부분 수정

관련된 추가적인 개발관련한 문의사항과 고급설정 등 관련 내용은
<http://adlibr.com> 의 개발자 지원 메뉴를 통해 공유되고 있습니다.

감사합니다.