

admixer

iOS SDK User Guide

2020.01

목차

- I. 개요
- II. **SDK 구성**
- III. 프로젝트 설정
- IV. 모바일 광고 초기화
- V. **Banner** 광고 추가
- VI. **Interstitial** 광고 추가
- VII. **Ad Network**별 추가 광고정보 설정
- VIII. 자주하는 질문

I. 개요

- 본 문서는 **AdMixer iOS SDK**를 프로젝트에 적용하기 위한 문서입니다.
- **AdMixer**는 광고 플랫폼을 통합하여 배너 및 전면 광고를 표시하는 기능을 제공합니다.
- 적용 버전은 다음과 같으며, 각각의 라이브러리는 해당 사이트에서 최신 버전을 받아 라이브러리를 교체하여 사용하실 수 있습니다. (제공되는 **AdMixer SDK**의 **light** 버전은 **Ad Network**사의 **SDK**를 포함하고 있지 않으나, **full**버전은 **lipo**를 통해 **simulator**용과 **release**용을 합친 **Ad Network** 사의 **SDK**가 포함되어 있습니다.)
 - Adfit (3.0.9)
 - Admob (7.53.1)
 - Cauly (3.1.7)
 - Dawin Click (T-AD, Syrup Ad) (3.7.0)
 - Facebook – Audience Network (5.6.0)
 - MANPLUS (MAN) (110)
 - Mopub (5.10.0)
 - Smaato (21.2.2)

I. 개요

➤ **AdMixer** 퍼블리셔 사이트를 통해 미디어에 대한 정보를 등록하여 **media key**를 발급 받으셔야 합니다.

미디어 하위에는 광고단위인 **adunit**을 생성해야 하며, **adunit** 생성 시 선택하신 사이즈와 **fullscreen** 여부에 맞게 광고 객체의 종류와 영역사이즈를 설정하셔야 합니다.

Fullscreen 값을 미설정하시면 **banner**객체로, 설정하시면 **interstitial** 객체로 사용이 가능합니다.

- **AdMixer**는 적용할 광고 플랫폼을 **AdMixer** 사이트를 통해 손쉽게 변경하실 수 있으며 각 광고 별 노출 비율 및 세부 설정 또한 사이트를 통해 변경하실 수 있습니다.
- ‘**Admixer**’ 네트워크를 이용하시면 서버 연동된 다수의 **dsp**로부터 다양한 광고를 수신 받을 수 있습니다. 이용에 참고 부탁드립니다.
- 띠배너와 전면배너 이외의 타 애드네트워크 광고를 추가하고자 하는 경우, 해당 애드네트워크 **SDK** 가이드를 참고하여 적용해 주시기 바랍니다.

II. SDK 구성 (1 / 3)

➤ **AdMixer iOS SDK**는 다음과 같이 구성되어 있습니다.

- Include : AdMixer iOS SDK를 control하기 위한 라이브러리 및 해당 header 파일
 - libAdMixer.a (AdMixer 라이브러리, 디바이스 빌드 전용)
 - AdMixer.h
 - AdMixerAdAdapter.h
 - AdMixerInfo.h
 - AdMixerInterstitial.h
 - AdMixerInterstitialPopupOption.h
 - AdMixerView.h
 - AXError.h
 - AXLog.h

II. SDK 구성 (2 / 3)

➤ **AdMixer iOS SDK**는 다음과 같이 구성되어 있습니다.

- libAdMixer_Combo.a
 - AdMixer 라이브러리로 디바이스/시뮬레이터 공용
 - libAdMixer.a 파일과 유사하나 시뮬레이터 빌드에서도 동작합니다.
 - 이 라이브러리를 사용하시려면 Include의 libAdMixer.a 파일 대신 이 파일을 프로젝트에 포함하시면 됩니다.
 - 이 라이브러리로 빌드 시에는 앱의 크기가 1~2MB 정도 커질 수 있습니다.

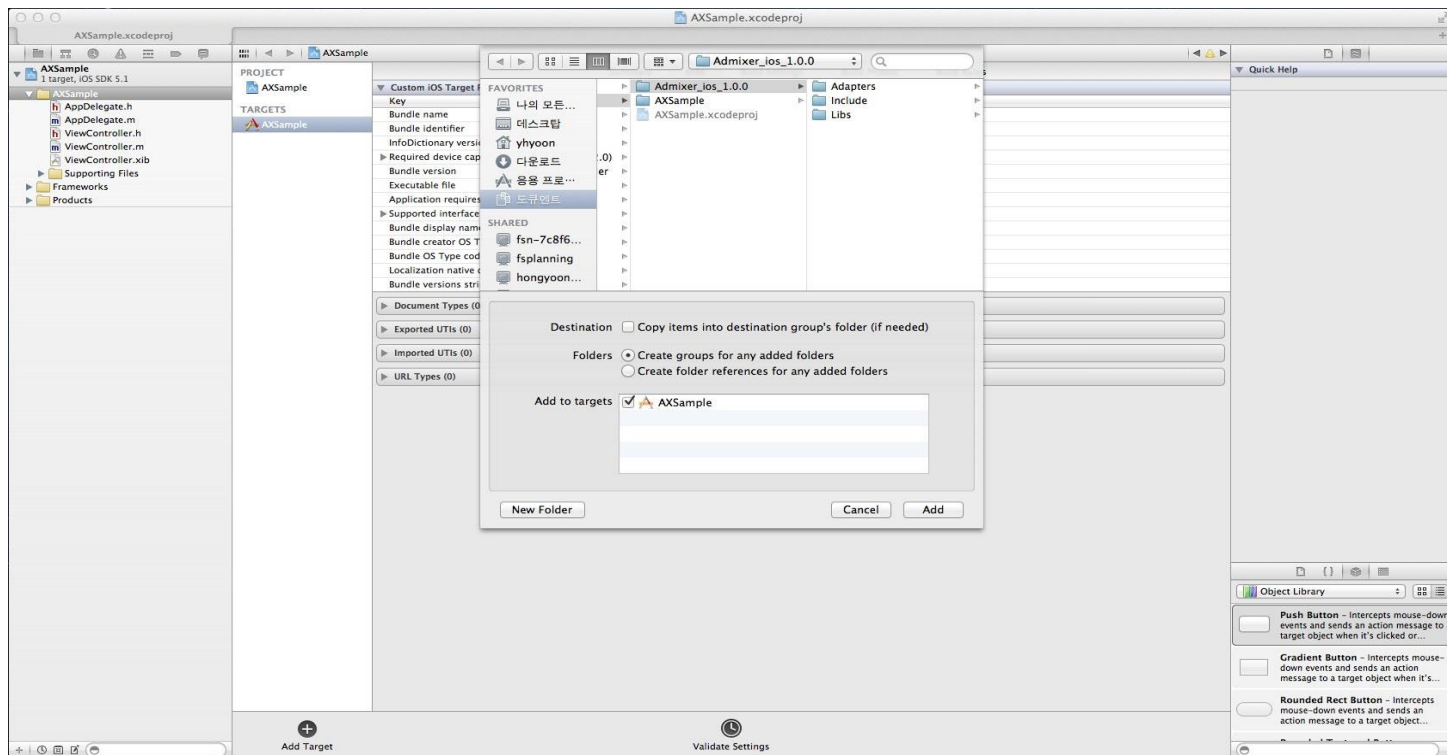
II. SDK 구성 (3 / 3)

➤ **AdMixer iOS SDK**는 다음과 같이 구성되어 있습니다.

- Adapters: 포함시키고자 하는 Ad Network의 adapter를 선별하시어 프로젝트에 포함시키시면 됩니다.
 - AdfitAdapter
 - AdmobAdapter
 - CautyAdapter
 - DawinClickAdapter
 - FacebookAdapter
 - ManAdapter
 - MopubAdapter
 - SmaatoAdapter
- AdMixerSample: AdMixer iOS SDK 사용 예제

III. 프로젝트 설정 (1 / 4)

1. **Admixer_ios_x.y.z.zip**을 풉니다.
2. **Include** 폴더를 프로젝트에 추가합니다. (**AdMixer** 라이브러리 추가)
3. **Adapters** 폴더를 프로젝트에 추가합니다. (**Adapter** 추가)



III. 프로젝트 설정 (2 / 4)

4. iOS 9 업데이트에 따른 추가 설정

- 프로젝트 설정의 Build Settings에서 Enable Bitcode를 NO로 설정해야 합니다.

- ATS(App Transport Security) 처리

: iOS9부터 애플 보안 정책에 따라 암호화 되지 않은 HTTP 통신이 차단되어 광고에 영향을 줄 수 있습니다.

따라서, 프로젝트 내에 프로젝트명-info.plist 파일에 아래 내용을 추가해주시기 바랍니다.

1) NSAppTransportSecurity(Type: Dictionary) 항목을 생성

2) 하위에 NSAllowsArbitraryLoads(Type: Boolean) 아이টে를 생성하여 YES로 설정합니다.

해당 설정을 통해 앱 내 모든 HTTP 통신을 허용할 수 있습니다. (For iOS 9 and later)

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key>
  <true/>
</dict>
```

**** (다음 장을 반드시 확인해주세요) ****

III. 프로젝트 설정 (3 / 4)

5. 2017년 iOS 10 업데이트에 따른 ATS 추가 설정

- ATS(App Transport Security) 처리

: 2017년부터 ATS를 활성화하도록 보안이 강화되었습니다. 따라서, 기존 설정에 추가적인 옵션값이 필요합니다.

(단, Ad Network에 따라 HTTPS 지원현황에 차이가 있을 수 있으므로, 반드시 각 Ad Network 가이드 내 ATS 설정값을 확인바랍니다.)

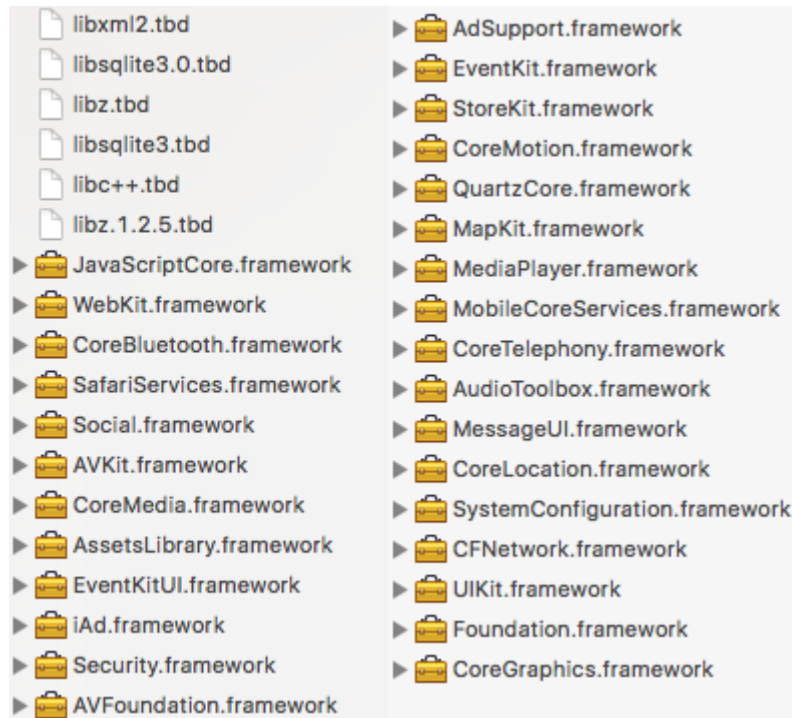
- 1) 기존 설정대로 NSAllowsArbitraryLoads 아이템을 YES로 설정합니다. (For iOS 9 and later)
- 2) NSAllowsArbitraryLoadsForMedia(Type: Boolean) 아이템을 생성하여 YES로 설정합니다. (For iOS 10)
- 3) NSAllowsArbitraryLoadsInWebContent(Type: Boolean) 아이템을 생성하여 YES로 설정합니다.

(admixer 배너 사용 시 권장하는 설정값이나, 웹뷰에서만 HTTP를 허용하는 값이므로 사용하시는 Ad Network에 따른 설정이 필요한 부분입니다.)

```
<key>NSAppTransportSecurity</key>
<dict>
  <key>NSAllowsArbitraryLoads</key> (해당 값만 설정 시, 앱 심사요청 과정에서 ATS 비활성 사유를 입력하셔야 합니다.)
  <true/>
  <key>NSAllowsArbitraryLoadsForMedia</key> (optional)
  <true/>
  <key>NSAllowsArbitraryLoadsInWebContent</key> (optional)
  <true/>
</dict>
```

III. 프로젝트 설정 (4 / 4)

- 아래는 광고 플랫폼 별로 필요한 설정을 하나로 묶어 놓은 것입니다. 다음의 **framework**이 포함되어 있는지 확인하시기 바랍니다.



- 자세한 내용은 해당 **Ad Network**사가 배포하는 기술문서를 참고하시기 바랍니다.

IV. 모바일 광고 초기화

➤ 아래 코드는 모바일 광고를 시작하기 위한 초기화 예제입니다.

- AdMixer 객체를 통해 반드시 1회 초기화 호출이 필요합니다.
- AdMixer 객체를 통해 필요한 adapter들을 등록해야 합니다.

```
- (BOOL)application(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {  
    // 로그 레벨 설정  
    [AdMixer setLogLevel:AXLogLevelDebug];  
  
    // 필요한 adapter 등록  
    [AdMixer registerAdapter:ADAPTER_ADFIT cls:[AdfitAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_ADMOB cls:[AdmobAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_CAULY cls:[CaulyAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_DAWIN_CLICK cls:[DawinClickAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_FACEBOOK cls:[FacebookAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_MAN cls:[ManAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_MOPUB cls:[MopubAdapter class]];  
    [AdMixer registerAdapter:ADAPTER_SMAATO cls:[SmaatoAdapter class]];  
  
    // AdMixer 초기화를 위해 반드시 광고호출 전에 앱에서 1회 호출해주셔야 합니다.  
    // adunits 파라미터는 앱 내에서 사용할 모든 adunit id를 배열형태로 넘겨주셔야 합니다.  
    [AdMixer initWithMediaKey:mediaKey adunits:adunits];  
  
    // COPPA(아동보호법) 관련 항목 설정값 - 선택사항  
    [AdMixer setTagForChildDirectedTreatment:AX_TAG_FOR_CHILD_DIRECTED_TREATMENT_TRUE];  
  
    ..... (계속) .....
```

IV. 모바일 광고 초기화

➤ 아래 코드는 모바일 광고를 시작하기 위한 초기화 예제입니다. (계속)

- AdMixer 객체를 통해 반드시 1회 초기화 호출이 필요합니다.
- AdMixer 객체를 통해 필요한 adapter들을 등록해야 합니다.

```
// Admob 적용 시에 SDK 초기화 호출이 필요
[[GADMobileAds sharedInstance] startWithCompletionHandler:nil];

// Mopub 적용 시에 SDK 초기화 호출이 필요
MPMoPubConfiguration *sdkConfig = [[MPMoPubConfiguration alloc]
                                     initWithAdUnitIdForAppInitialization:@"my_any_mopub_adunit_id"];
[[MoPub sharedInstance] initializeSdkWithConfiguration:sdkConfig completion:nil];

// Smaato 적용 시에 SDK 초기화 호출이 필요
SMAConfiguration *config = [[SMAConfiguration alloc] initWithPublisherId:@"my_smaato_publisher_id"];
[SmaatoSDK initSDKWithConfig:config];

}
```

V. Banner 광고 추가 – 광고 뷰 추가

- 아래 코드는 **Banner** 광고를 추가한 예제입니다.
- Banner광고를 위해서는 AdMixerViewDelegate 구현하셔야 합니다.
 - Banner광고를 담는 AdMixerView가 필요합니다.

```
@interface ViewController : UIViewController<AdMixerViewDelegate, AdMixerInterstitialDelegate> {  
    AdMixerView * _adView;  
}
```

V. Banner 광고 추가 – 광고 뷰 추가 (계속)

➤ 아래 코드는 **Banner** 광고를 추가한 예제입니다.

- 광고정보를 관리하는 객체(AdMixerInfo)를 생성하고, adunitID필드를 발급받은 adunit id로 채웁니다.
- 광고영역의 사이즈는 반드시 생성하신 adunit에 맞게 설정하시기 바랍니다.

```
- (void) viewDidLoad
{
    [super viewDidLoad];

    AdMixerInfo * adInfo = [[[AdMixerInfo alloc] init] autorelease];
    adInfo.adunitID = @"my_adunit_id"; // adunit id값을 설정합니다.

    _adView = [[AdMixerView alloc] initWithFrame:CGRectMake(0, 0, 320, 50)];
    _adView.center = CGPointMake(self.view.center.x, _adView.center.y);
    // 320*50사이즈 일때만, 가로를 디바이스에 맞추거나 영역사이즈로 설정하는 것 중에 선택가능
    // 320*50을 제외한 사이즈일때는 영역사이즈로 설정해야 함

    // 각 애드네트워크별 광고설정값을 아래와 같이 설정할 수 있으며,
    // adunit 사이즈에 맞게 네트워크 사이즈를 설정하시기 바랍니다. (자세한 사항은 Ad Network별 추가 광고정보 설정 참고)
    [adInfo setAdapterAdInfo:ADAPTER_ADMOB infoKey:@"adSize" withStringInfoValue:@"kGADAdSizeBanner"];
    [adInfo setAdapterAdInfo:ADAPTER_FACEBOOK infoKey:@"adSize" withStringInfoValue:@"kFBAdSizeHeight50Banner"];
    // Man의 경우, 광고요청시 필수 파라미터로 사용될
    // appID, appName, storeURL 정보를 아래와 같이 꼭 따로 설정해주셔야 정상동작합니다.
    [adInfo setAdapterAdInfo:ADAPTER_MAN infoKey:@"appID" withStringInfoValue:@"앱스토어 appID"];
    [adInfo setAdapterAdInfo:ADAPTER_MAN infoKey:@"appName" withStringInfoValue:@"앱명 - 영문으로 기입"];
    [adInfo setAdapterAdInfo:ADAPTER_MAN infoKey:@"storeURL" withStringInfoValue:@"앱의 실제 store url"];
    ..... (계속) .....
```

V. Banner 광고 추가 – 광고 뷰 추가 (계속)

➤ 아래 코드는 **Banner** 광고를 추가한 예제입니다. (계속)

- 광고정보를 관리하는 객체(AdMixerInfo)를 생성하고, adunitID필드를 발급받은 adunit id로 채웁니다.
- 광고영역의 사이즈는 반드시 생성하신 adunit에 맞게 설정하시기 바랍니다.

```
_adView.delegate = self;
[self.view addSubview:_adView];

[_adView startWithAdInfo:adInfo baseViewController:self];

// 앱 백그라운드 진입 시 혹은 다른 뷰로 넘어갈 때 띠배너 객체를 해제하는 방법은 아래와 같습니다.
[_adView removeFromSuperview];
_adView = nil;
}
```


V. Banner 광고 추가 – 광고 뷰 추가 (계속)

➤ AdMixerViewDelegate를 통한 이벤트 수신

- AdMixerViewDelegate를 위한 method를 구현합니다.
- onSucceededToReceiveAd는 광고를 잘 받아온 경우 호출됩니다.
- onFailedToReceiveAd는 광고 받기에 실패한 경우 호출됩니다.
- 현재 선택된 adapter 이름 확인 : [adView currentAdapterName];

```
#pragma mark - AdMixerViewDelegate

- (void)onSucceededToReceiveAd:(AdMixerView *)adView {
    NSLog(@"onSucceededToReceiveAd");
}

- (void)onFailedToReceiveAd:(AdMixerView *)adView error:(AXError *)error {
    NSLog(@"onFailedToReceiveAd : %@", error.errorMsg);
}
```

VI. Interstitial 광고 추가 – 광고 형태

➤ 전면광고에는 2가지 형태가 제공됩니다.

- 일반 전면광고
- 팝업형 전면광고 (admixer ad network만 사용 가능합니다.)

일반 전면광고 예시



팝업형 전면광고 예시



VI. Interstitial 광고 추가 – 광고 뷰 추가

➤ 아래 코드는 **Interstitial** 광고를 추가한 예제입니다.

- Interstitial 광고를 위해서는 `AdMixerInterstitialDelegate` 구현하셔야 합니다.
- Interstitial 광고는 1회성 객체입니다. `start` 혹은 `load` 메소드 호출은 한 번만 가능합니다.
- `startWithAdInfo` API를 호출하시면 `onSucceededToReceiveInterstitialAd` 이벤트와 동시에 자동적으로 전면 광고가 표시됩니다.
- `loadWithAdInfo` API를 호출하시면 `onSucceededToReceiveInterstitialAd` 이벤트를 받아서 광고 로딩이 성공하면 원하는 시점에 `displayAd` API로 전면 광고를 노출하게 됩니다.
- `loadWithAdInfo` 지원 Ad Network : Admob, Cauly, Dawin Click(SyrupAd, T-ad), Facebook, Mopub (다른 AdNetwork는 `loadWithAdInfo` 호출 시 로드하지 않습니다.)
- `loadWithAdInfo` API를 호출하고 일정시간이 지나면, 광고가 노출이 되어도 유효노출로 처리되지 않는 경우가 있습니다. 이 경우는 **ad network**별로 다르므로, 해당 네트워크 사에 확인하여 `loadWithAdInfo` 호출 후 적절한 타임아웃을 걸어서 재호출 하는 방식으로 사용하시기 바랍니다. 최소호출간격이 있는 경우도 있으므로, 적당한 타임아웃을 설정하시기 바랍니다. (ex. AdMob 광고노출인증유효시간 20분)

VI. Interstitial 광고 추가 – 광고 뷰 추가

➤ 아래 코드는 Interstitial광고를 추가한 예제입니다. (계속)

```
@property(nonatomic, strong) AdMixerInterstitial * interstitial;
- (IBAction)interstitialAdButtonAction:(id)sender {

    AdMixerInfo * adInfo = [[[AdMixerInfo alloc] init] autorelease];
    adInfo.adunitID = @"my_adunit_id"; // adunit id값을 설정합니다.
    // adInfo.interstitialTimeout = 20; // 전면 요청 timeout 설정(Default값은 0으로 서버에서 주는 시간으로 처리)
    // adInfo.setBackgroundAlpha = YES; // 전면 노출 시 광고 외 영역 반투명 처리(Default값은 NO)

    // 팝업형 전면광고 설정코드 (선택사항)
    AdMixerInterstitialPopupOption *adConfig = [[[AdMixerInterstitialPopupOption alloc] init]; // 팝업형 전면광고 옵션설정
    [adConfig setButton:@"광고종료" withButtonColor:[UIColor whiteColor]];
    // 광고를 닫는 기능의 버튼이 제공되며, 버튼문구와 버튼색상을 지정가능 (버튼문구 Default값은 '광고종료')
    [adConfig setButtonFrameColor:[UIColor blackColor]]; // 버튼프레임 색상을 지정가능
    [adInfo setInterstitialAdType:AdMixerInterstitialPopupType withInterstitialPopupOption:adConfig];
    // 팝업형 전면광고 option을 adInfo에 적용 (option nil로 설정시 Default option으로 제공됨)
    // AdMixerInterstitialBasicType : 일반(Default), AdMixerInterstitialPopupType : 팝업형

    // 전면광고는 1회성 객체이며,
    // adunit_id와 1:1 매핑되므로 광고호출시 자원이 남아있으면 명시적으로 해제함
    [self stopInterstitial];

    interstitial = [[[AdMixerInterstitial alloc] init];
    interstitial.delegate = self;
    [interstitial startWithAdInfo:adInfo baseViewController:self]; // 광고 로딩 시 바로 표시
    // [interstitial loadWithAdInfo:adInfo baseViewController:self]; // 광고 로딩 후 원하는 시점에 displayAd를 호출
}
```

VI. Interstitial 광고 추가 – 광고 뷰 추가

➤ 아래 코드는 **Interstitial** 광고를 추가한 예제입니다. (계속)

```
- (void)stopInterstitial {  
    if(interstitial != nil) {  
        [interstitial stop];  
        interstitial = nil;  
    }  
}
```

- AdMixerInterstitialDelegate를 위한 method를 구현합니다.
- onSucceededToReceiveInterstitialAd는 광고를 잘 받아온 경우 호출됩니다.
- onFailedToReceiveInterstitialAd는 광고 받기에 실패한 경우 호출됩니다.
- onCloseInterstitialAd는 전면 광고 창이 닫혔을 때에 호출됩니다.
- onDisplayedInterstitialAd는 전면 광고가 화면에 보여졌을 때에 호출됩니다. (선택사항)
- onClickedPopupButton은 팝업형 전면광고 버튼을 누르면 호출됩니다. (선택사항)

VI. Interstitial 광고 추가 – 광고 뷰 추가

➤ 아래 코드는 **Interstitial**광고를 추가한 예제입니다. (계속)

```
#pragma mark - AdMixerInterstitialDelegate

- (void)onSucceededToReceiveInterstitialAd:(AdMixerInterstitial *)interstitial {
    NSLog(@"onSucceededToReceiveInterstitialAd");
}

- (void)onFailedToReceiveInterstitialAd:(AdMixerInterstitial *)interstitial error:(AXError *)error {
    NSLog(@"onFailedToReceiveInterstitialAd : %@", error.errorMsg);
    [self stopInterstitial];
}

- (void)onClosedInterstitialAd:(AdMixerInterstitial *)interstitial {
    NSLog(@"onClosedInterstitialAd");
    [self stopInterstitial];
}

- (void)onDisplayedInterstitialAd:(AdMixerInterstitial *)interstitial {
    NSLog(@"onDisplayedInterstitialAd");
}

- (void)onClickedPopupButton:(AdMixerInterstitial *)interstitial {
    NSLog(@"onClickedPopupButton");
}
```

VI. Interstitial 광고 추가 – 닫기 (close)

➤ 아래 코드는 **Interstitial**광고 노출 후 원하는 시점에서 닫기 위해 **close method**를 구현한 예입니다.

- 전면광고 close지원 Ad Network : Admixer, Housead, Dawin Click
- 전면광고가 화면에 노출되기 전에 close를 호출하게 되면 동작하지 않습니다.
- 아래와 같이 타이머를 사용할 경우, 반드시 **close delegate**에서 타이머를 해제하여야 합니다. (타이머 작동 이전에 사용자가 전면광고를 끄게 되면, close method가 다음 전면광고에 영향을 줄 수 있습니다.)

```
AdMixerInterstitial *_interstitial ;
NSTimer *timer;
- (void)onDisplayedInterstitialAd:(AdMixerInterstitial *)intersitial {
    NSLog(@"onDisplayedInterstitialAd");

    timer = [NSTimer scheduledTimerWithTimeInterval:5 target:self
        selector:@selector(closeInterstitial) userInfo:nil repeats:NO];
    // 전면광고 노출 후 5초 타이머 설정
}
- (void)onClosedInterstitialAd:(AdMixerInterstitial *)intersitital {
    NSLog(@"onClosedInterstitialAd");
    [timer invalidate]; // 타이머 해제
}
- (void)closeInterstitial {
    if(_interstitial) {
        [_interstitial close]; // 전면광고 닫기
        _interstitial = nil;
    }
}
```

VII. Ad Network별 추가 광고정보 설정

➤ **setAdapterAdInfo API는 AdNetwork 별로 제공되는 옵션을 설정하기 위한 API입니다.**

➤ **지원 옵션**

- Admob

```
[adInfo setAdapterAdInfo:ADAPTER_ADMOB infoKey:@"adSize"  
withStringInfoValue:@"kGADAdSizeSmartBannerPortrait"];
```

- Admob 배너 종류 설정

kGADAdSizeSmartBannerPortrait, kGADAdSizeSmartBannerLandscape,
kGADAdSizeBanner, kGADAdSizeMediumRectangle, kGADAdSizeFullBanner,
kGADAdSizeLeaderboard

- Default : kGADAdSizeSmartBannerPortrait

- Cauly

```
[adInfo setAdapterAdInfo:ADAPTER_CAULY infoKey:@"adSize"  
withStringInfoValue:@"CaulyAdSize_IPhone"];
```

- Cauly 배너 종류 설정

CaulyAdSize_IPhone, CaulyAdSize_IPadSmall, CaulyAdSize_IPadLarge

- Default : CaulyAdSize_IPhone

VII. Ad Network별 추가 광고정보 설정

➤ **setAdapterAdInfo API는 AdNetwork 별로 제공되는 옵션을 설정하기 위한 API입니다.**

➤ **지원 옵션 (계속)**

- Man

```
[adInfo setAdapterAdInfo:ADAPTER_MAN infoKey:@"appID" withStringInfoValue:@""]; // 필수
```

- Man 배너 호출 시, 파라미터 appID(=앱스토어 appID)를 설정

- Default : ""

```
[adInfo setAdapterAdInfo:ADAPTER_MAN infoKey:@"appName" withStringInfoValue:@""]; // 필수
```

- Man 배너 호출 시, 파라미터 appName(=앱명 – 영문으로 기입)를 설정

- Default : ""

```
[adInfo setAdapterAdInfo:ADAPTER_MAN infoKey:@"storeURL" withStringInfoValue:@""]; // 필수
```

- Man 배너 호출 시, 파라미터 storeURL(=앱의 실제 store url)를 설정

- Default : ""

VII. Ad Network별 추가 광고정보 설정

➤ **setAdapterAdInfo API**는 **AdNetwork** 별로 제공되는 옵션을 설정하기 위한 **API**입니다.

➤ 지원 옵션 (계속)

- Facebook

```
[adInfo setAdapterAdInfo:ADAPTER_FACEBOOK infoKey:@"adSize"  
withStringInfoValue:@"kFBAdSizeHeight50Banner"];
```

- Facebook 배너 종류 설정

kFBAdSizeHeight50Banner, kFBAdSizeHeight90Banner,
kFBAdSizeHeight250Rectangle

- Default : kFBAdSizeHeight50Banner

VII. 자주하는 질문 (1 / 3)

➤ 문제 확인을 위한 로그는 없나요?

- [AdMixer setLogLevel:AXLogLevelAll];
- 위의 코드를 넣으시면 상세한 로그를 확인하실 수 있습니다. 문제 발생 시 해당 로그를 전달해 주시면 좀 더 정확한 원인 파악에 도움이 됩니다.

➤ 하나의 App에 복수 개의 Media Key를 적용해도 되나요?

- 한 개의 App에는 한 개의 Media Key만 적용하셔야 합니다.

➤ 동일한 Adunit ID로 여러 개의 광고객체를 생성해도 되나요?

- 한 개의 Adunit ID는 한 개의 광고객체에서만 사용가능합니다.

➤ 광고객체를 정상적으로 호출했으나 광고요청이 이루어지지 않습니다.

- AdMixer 객체를 통해 초기화 호출을 해주셨는지 확인 부탁드립니다. 초기화 시에 설정한 Media Key 및 adunit ID 배열과 일치하는 정보를 가진 광고객체만 정상적으로 동작합니다.
- Adunit 생성 시에 입력한 fullscreen 정보에 따라서 사용가능한 광고객체가 다릅니다. banner 객체는 fullscreen이 off 일 때, interstitial객체는 fullscreen이 on일 때 각각 사용할 수 있습니다.

➤ Adunit에 설정한 사이즈와 다른 사이즈의 광고가 노출됩니다.

- Adunit에 설정한 사이즈값은 admixer와 housead의 경우는 내부적으로 사이즈가 보장되지만, 타 Ad Network사의 경우, 해당 사이즈에 맞는 지면을 생성하셔서 전략에 키값을 설정하시고, 사이즈옵션이 있다면 코드상에서 setAdapterAdInfo를 통해 개별적으로 설정하셔야 정상 노출이 보장됩니다.

VII. 자주하는 질문 (2 / 3)

➤ 한 App내에서 많은 adunit을 사용해도 되나요?

- 사용 가능한 adunit수에 제한은 없지만, 광고 객체를 설정하고 로딩하는데에 많은 메모리가 할당되기 때문에 앱을 성능을 위해서 많은 광고객체 호출은 지양하시는 것이 좋습니다.

➤ 시뮬레이터 Configuration에서는 빌드가 되지 않습니다.

- 기본으로 포함되어 있는 libAdMixer.a 라이브러리는 디바이스 전용 라이브러리입니다. 상위 폴더에 있는 libAdMixer_Combo.a 라이브러리로 교체하시면 됩니다.

➤ ARC(Automatic Reference Counting) 설정이 되어 있지 않은 것 같습니다.

- ARC는 빌드 시에 소스파일(.m)에 적용하는 옵션으로 라이브러리 자체에는 ARC 적용에 따른 차이는 없습니다. ARC가 적용된 프로젝트 혹은 소스에서는 AdMixer 관련 Class 사용 시 retain, release를 호출하실 필요가 없고, ARC가 적용되지 않은 소스에서는 반드시 retain, release를 적절하게 호출해 주셔야 합니다. 다만, ARC 프로젝트 내에서 각 adapter(.m)파일을 import하여 사용하실 때는 build phases에서 compile sources에 있는 adapter(.m)파일마다 -fno-objc-arc 옵션값을 설정하시기 바랍니다.

➤ 라이브러리 크기를 줄일 수 있습니까?

- 모든 광고를 적용하시는 편이 표시할 광고가 없는 상황을 줄일 수 있기 때문에 가급적이면 모든 광고를 포함하시는 편이 좋겠지만 프로그램 크기가 커져서 문제가 되신다면 꼭 필요한 Ad Network를 결정하시고 불필요한 adapter를 삭제하시고, 필요한 Ad Network adapter만 등록하시면 됩니다.

VII. 자주하는 질문 (3 / 3)

➤ 전면 광고가 자주 나오지 않습니다.

- 전면 광고의 경우 배너 광고보다 광고 물량이 적어 설정하신 Ad Network가 적은 경우 광고가 나오지 않을 확률도 그 만큼 커집니다.

The End

admixer

<http://www.admixer.co.kr>