



SDK v3.2 Integration Quickstart

iOS 8 and Higher

Ampiri iOS SDK Integration

This document is intended as a basic integration of the Ampiri SDK for iOS. If you require additional information, please consult our [SDK Integration Support](#) or the [Readme](#) file.

OPTION 1: Integrate the SDK via Cocoapods

- A. To integrate the Ampiri SDK with ALL mediated ad networks, add `pod 'ampiri-ios-sdk'` to your podfile.
- B. To integrate the Ampiri SDK with server-side ad networks, add `pod 'ampiri-ios-sdk/Core'` to your podfile.

Choosing “option B” allows you to select the client-side ad networks you want to use with Ampiri:

- **Facebook Audience** `ampiri-ios-sdk/Adapters/FBAudienceAdapter`
- **Google Mobile Ads SDK** `ampiri-ios-sdk/Adapters/GoogleMobileAdsAdapter`
- **MoPub** `ampiri-ios-sdk/Adapters/MopubAdapter`
- **Chartboost** `ampiri-ios-sdk/Adapters/ChartboostAdapter`
- **NativeX** `ampiri-ios-sdk/Adapters/NativeXAdapter`
- **Unity Ads** `ampiri-ios-sdk/Adapters/UnityAdsAdapter`
- **Vungle** `ampiri-ios-sdk/Adapters/VungleAdapter`
- **AdColony** `ampiri-ios-sdk/Adapters/AdColonyAdapter`
- **AppLovin** `ampiri-ios-sdk/Adapters/AppLovinAdapter`
- **Baidu** `ampiri-ios-sdk/Adapters/BaiduAdapter`

NOTE: Baidu libraries and adapters can ONLY be integrated manually. They cannot be integrated with the Ampiri SDK via Cocoapods.

OPTION 2: Integrate the SDK manually

Step 1 - Download the SDK

1. [Download](#) the Ampiri SDK
2. Add `AmpiriSDK.framework` to your project
3. Add `AMPVastLib.framework`
4. In Xcode, add `-ObjC` to the *project settings* -> *build settings* tab. [How-to?](#)

Step 2 - Add Ad Network Libraries

Ampiri has been **tested** to run with the following 3rd-party SDKs:

- AdColony 2.6.2
- Chartboost 6.2.1
- Google Mobile Ads 7.12.0
- NativeX 5.5.7.1
- Unity Ads 1.5.6
- Vungle 3.2.2
- Facebook Audience 4.16.0
- MoPub 4.9.1
- AppLovin 3.1
- Baidu 4.3.1

NOTE: *Integrating Baidu and MoPub within a single application results in errors.*

Banner Libraries

- [Google](#)
- [MoPub](#)
- [Facebook](#)

Interstitial Libraries

- [Google](#)
- [Chartboost](#)
- [Facebook](#)
- [MoPub](#)
- [NativeX](#)
- [AppLovin](#)

Video Libraries

- [AdColony](#)
- [UnityAds](#)
- [Chartboost](#)
- [NativeX](#)
- [Vungle](#)

Native Libraries

- [Google](#)
- [Facebook](#)
- [MoPub](#)
- [AppLovin](#)
- [Baidu](#)

Step 3 - Add the SDK Adapters

- | | |
|---------------------|------------------------------|
| • AdColony | libAMPAdColonyAdapter |
| • Chartboost | libAMPChartboostAdapter |
| • Google Mobile Ads | libAMPGoogleMobileAdsAdapter |
| • NativeX | libAMPNativeXAdapter |
| • Unity Ads | libAMPUnityAdsAdapter |
| • Vungle | libAMPVungleAdapter |
| • Facebook Audience | libAMPFBAudienceAdapter |
| • MoPub | libAMPMopubAdapter |
| • AppLovin | libAMPAppLovinAdapter |
| • Baidu | libAMPBaiduAdapter |

Step 4 - Required System Libraries

In *Xcode*, follow the steps below:

1. Go to *project settings*
2. Select *build settings*
3. Set the flag - **Link Frameworks automatically** to **YES**.
4. [How-to?](#)

Step 5 - ATS Settings

NOTE: These settings are valid for iOS 9.0 and higher.

It is **recommended** that the AppTransportSecurity feature in iOS 9 (or higher) be **disabled** to ensure correct functioning of the Ampiri SDK. In your application **Info.plist** file, add the following settings:

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

Step 6 - Ad Types

Set **testMode** to **YES** in the following manner (for *AdMob*, *Facebook*, *Unity* implementation):

```
#import <AmpiriSDK/AmpiriSDK.h>
[AmpiriSDK setTestMode:YES];
```

Banners

- Advertising space ID for Banner testing: "04c447d7-ffb8-4ba1-985e-4d2b9f88cd69"
- Sizes: 320x50 or 728x90
- UIViewController

Use the following methods in your **UIViewController** subclass:

```
- (AMPBannerView *)loadBannerWithSize:(CGSize)size
adUnitId:(NSString *)adUnitId
success:(void (^)(AMPBannerView *banner))success
failure:(void (^)(AMPError *error))failure;
```

Add following code to **viewDidAppear** and **viewWillDisappear** methods:

Objective-c:

viewDidAppear and viewWillDisappear

```
-(void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];
    [self.bannerView resumeAutoUpdate];
}
```

```
-(void)viewWillDisappear:(BOOL)animated {
    [super viewWillDisappear:animated];
    [self.bannerView pauseAutoUpdate];
}
```

Swift:

viewDidAppear and viewWillDisappear

```
override func viewDidAppear(animated: Bool) {  
    self.bannerView?.resumeAutoUpdate()  
}
```

```
override func viewDidDisappear(animated: Bool) {  
    self.bannerView?.pauseAutoUpdate()  
}
```

Banner Events

- (void (^)(AMPBannerView *banner))success
- (void (^)(AMPError *error))failure

Interstitial

- Advertising space ID for Interstitials testing: "2cb34a73-0012-4264-9526-bde1fce2ba92"

Use the following methods in your `UIViewController` subclass:

```
- (void)loadFullscreenWithAdUnitId:(NSString *)adUnitId  
    options:(AMPFullscreenLoadOptions)options  
    viewController:(UIViewController *)viewController  
    success:(void (^)(AMPFullscreenBannerController *fullscreenController))success  
    failure:(void (^)(AMPError *error))failure;
```

Interstitial Presentation

```
typedef NSInteger(NSInteger, AMPFullscreenLoadOptions)  
{  
    AMPFullscreenLoadOptionsDefault = 0,  
};
```

Interstitial Events

- (void (^)(AMPFullscreenBannerController *fullscreenController))success
- (void (^)(AMPError *error))failure

Video

- Advertising space ID for Banners testing: "87f65c4c-f12d-4bb6-96fd-063fe30c4d69"

Use the following methods in your `UIViewController` subclass:

```
- (void)loadVideoWithAdUnitId:(NSString *)adUnitId
  success:(void (^)(AMPVideoController *videoController))success
  failure:(void (^)(AMPErrror *error))failure;
```

Video Events

- (void (^)(AMPVideoController *videoController))success
- (void (^)(AMPErrror *error))failure

Native

- Advertising space ID for Banners testing: "7f900c7d-7ce3-4190-8e93-310053e70ca2"
- Sizes: 320x50 or 728x90

Use the following methods in your `UIViewController` subclass:

```
- (void)loadNativeAdWithAdUnitId:(NSString *)adUnitId
  parentViewController:(UIViewController *)viewController
  adViewClassForRendering:(Class)adViewClass
  success:(void (^)(UIView *adNativeViewContainer))success
  failure:(void (^)(AMPErrror *error))failure;
```

When implementing a native ad use the following common algorithm:

1. Create your own subclass of `AMPNative` with any name such as `MyNativeBannerView`
2. Perform the following:
 - `XIB` File – Design the layout of `MyNativeBannerView` in a separate `XIB` file. Bind the UI controls in this `XIB` and properties from `<AMPNativeViewInterface>`, which is adopted by `AMPNativeView`. The implementation of `MyNativeBannerView` class must override the `+(NSString*)xibName` method that returns the name of the same `XIB`.
3. Call `loadNativeAdWithSize` with the required parameters, where the `adUnitId` is your private advertising space ID and `className` is the name of the `MyNativeBannerView` class. After downloading the ad data, the SDK initiates an instance of the `MyNativeBannerView` class created in step 2. After that the SDK renders native ad data in bound controls of this instance. Only the following controls are filled in this manner (main): `ampTitleTextLabel`, `ampMainTextLabel`, `ampIconImageView`, `ampMainMediaView`. When rendering has completed, the *successful* completion block with this instance is invoked.
4. Render the other controls, like `ampRatingView`, in the `AdView` container.
5. Show ad view container on the screen.
6. Register `AdView` container for the taps handling

Location Control

Advertising space for testing is identical to native ads: "7f900c7d-7ce3-4190-8e93-310053e70ca2"

For a simple integration, you can use the templates In-Feed and Content Stream based on

`AMPNativeAdsTemplateType` enum:

```
- (AMPTableViewStreamAdapter *)addLocationControlToTableView:(UITableView *)tableView
    parentViewController:(UIViewController *)viewController
    adUnitId:(NSString *)adUnitId
    templateType:(AMPNativeAdsTemplateType)templateType
    templateCustomization:(void (^)(AMPTemplateCustomizationObject
*templateCustomizationObject))templateCustomization;
```

```
- (AMPCollectionViewStreamAdapter *)addLocationControlToCollectionView:(UICollectionView
*)collectionView
    parentViewController:(UIViewController *)viewController
    adUnitId:(NSString *)adUnitId
    templateType:(AMPNativeAdsTemplateType)templateType
    templateCustomization:(void (^)(AMPTemplateCustomizationObject
*templateCustomizationObject))templateCustomization;
```