



AdFalcon iOS SDK 1.0.4 Developer's Guide

AdFalcon Mobile Ad Network

Product of

Noqoush Mobile Media Group



Table of Contents

1	Introduction	3
	OS version support.....	3
	In the zip file.....	3
	Frameworks	3
2	iOS SDK Classes and Enum Description	4
	ADFAView Class.....	4
	ADFAViewDelegate Protocol.....	5
	ADFUserInfo Class	5
	ADFTargetingParams Class.....	6
	ADFAViewAdUnitSize Enum.....	7
	ADFAViewError Enum	7
3	AdFalcon iOS SDK Integration	9
	Step 1: Add headers and library.....	9
	Step 2: Add frameworks.....	9
	Step 3: Add snap of code to a header file	10
	Step 4: Add snap of codes to a source file	10
4	More Information:	13



1 Introduction

AdFalcon iOS SDK Integration guide contains all the information needed by an iOS developer to integrate with the AdFalcon Network. The guide will also show examples and code snippets on how to perform the integration.

OS version support

iOS SDK supports iPhone and iPad platforms utilizing iOS version 3.0 to the latest iOS version.

In the zip file

You will find the following files within the iOS SDK folder:

- ADFAdView.h
- ADFAdViewDelegate.h
- ADFUserInfo.h
- ADFTargetingParams
- libAdFalconSDK1.0.4_iOS30.a

Frameworks

You must add the following frameworks to enable iOS SDK to work:

- QuartzCore.framework
- MediaPlayer.framework
- CoreGraphics.framework
- SystemConfiguration.framework



2 iOS SDK Classes and Enum Description

ADFAView Class

This is the main class and it extends UIView. the Class is responsible for gathering all the needed information from a developer to get an ad from AdFalcon network and rendering it on the device.

The properties of the ADFAView Class are:

Property	Description
<code>int</code> refreshDuration	Refresh duration in seconds
<code>NSObject<ADFAViewDelegate> *</code> delegate	Delegate of AdFalcon view
<code>BOOL</code> testing	The property is used to inform AdFalcon network that the application is under the testing mode rather than production mode.
<code>BOOL</code> logging	Enable logging

The methods of the ADFAView Class are:

Method	Description
<pre> -(void) initializeWith AdUnit:(ADFAViewAdUnitSize) adUnit siteId: (NSString*) siteId userInfo: (ADFAUserInfo*) userInfo rootViewController: (UIViewController*) rootViewController enableAutorefresh: (BOOL) enableAutorefresh delegate: (NSObject<ADFAViewDelegate> *) delegate; </pre>	<p>Will initialize the AdFalcon view and load the ad.</p> <p>All the parameters are required except “delegate” which can be null.</p>
<pre> -(void) pauseAutoRefresh; </pre>	<p>Will pause auto refresh timer which is responsible for getting new ad the ad after a certain period has elapsed.</p>
<pre> -(void) resumeAutoRefresh; </pre>	<p>Will resume the auto refresh timer which responsible for getting new ad after a certain period has elapsed.</p>
<pre> -(void) refreshAd; </pre>	<p>Will get a new ad from AdFalcon.</p>



ADFAdViewDelegate Protocol

AdFalcon SDK provides a protocol called ADFAdViewDelegate that is responsible for providing feedback from AdFalcon SDK. This protocol contains nine optional delegate methods which are called upon the following events:

- When an error occurs
- Before and After an ad is loaded
- Before and After ad click action screen is displayed
- Before and After ad click action screen is closed
- Application will go to background mode and
- Application will terminate.

The methods of the ADFAdViewDelegate Class are:

Method	Description
<code>-(void)adViewWillLoadAd: (ADFAdView*) adView;</code>	Will be called before the ad is being loaded.
<code>-(void)adViewDidLoadAd: (ADFAdView*) adView;</code>	Will be called after the ad has been loaded
<code>-(void)adView: (ADFAdView*) adView didFailWithCode:(int) code message:(NSString*) message;</code>	Will be called when an error has occurred during loading an ad.
<code>-(void)adViewWillPresentScreen: (ADFAdView*) adView;</code>	Will be called before the ad click screen is displayed.
<code>-(void)adViewDidPresentScreen: (ADFAdView*) adView;</code>	Will be called after the ad click screen is displayed.
<code>-(void)adViewWillDismissScreen: (ADFAdView*) adView;</code>	Will be called before the ad click screen is dismissed.
<code>-(void)adViewDidDismissScreen: (ADFAdView*) adView;</code>	Will be called after the ad click screen is dismissed.
<code>-(void)applicationWillTerminate:(UIApplication *)application;</code>	Will be called before the Application is terminated.
<code>-(void)applicationDidEnterBackground:(UIApplication *)application;</code>	Will be called before the Application enters into background mode.

ADFUserInfo Class

The ADFUserInfo Class contains all the needed parameters about the user of application.

The parameters of the ADFUserInfo Class are:

Parameter	Required	Description	Values
Language	No	The language of the requested ad in ISO	ar, en



			639-1 language codes format (Two Letters code);	
Postal code	No		A parameter containing the user's postal/ZIP code	11121
Area code	No		A parameter containing the user's area code	06
Age	No		A parameter containing the user's age	27
Gender	No		A parameter containing the user's gender	kADFUserInfoGenderNone kADFUserInfoGenderMale kADFUserInfoGenderFemale
Country code	No		Country code of the end user in ISO 3166-1 alpha-2 format code (two-letter code)	JO, SA ...etc.
Birthdate	No		Birthdate of application user in format dd.MM.yyyy	21.11.1984
Location: Latitude, Longitude	No		The geolocation information of the device. The location information is divided into two double values; latitude and longitude.	35.658, 34.641

ADFTargetingParams Class

The ADFTargetingParams Class contains all the needed parameters about any given user and application in order to help adFalcon network to send most related and targeted ads to the user and application. All parameters are optional.

The parameters of the ADFTargetingParams Class are:

Parameter	Required	Description	Values
-----------	----------	-------------	--------



userInfo	No	A class containing information about the user of application	ADFUserInfo
Keywords	No	A list containing keywords in comma separated format. AdFalcon's ads selector engine will search for ads containing these keywords.	ex. sport, news, lifestyle, ...etc.
Additional Info	No	A map of keys and values to add additional parameters.	

ADFAdViewAdUnitSize Enum

ADSAdViewAdUnitSize Enum defines the ad units supported by the AdFalcon iOS SDK

Enum	Description	Size	Devices
kADFAdViewAdUnitSize320x48	Standard	320 x 48	All
kADFAdViewAdUnitSize300x250	Medium Rectangle	300 x 250	All
kADFAdViewAdUnitSize468x60	Full Banner	468 x 60	Tablet
kADFAdViewAdUnitSize728x90	Leaderboard	728 x 90	Tablet
kADFAdViewAdUnitSize120x600	Skyscraper	120 x 600	Tablet

ADFAdViewError Enum

Enum	Description
kADFAdViewErrorInternalServer	is an error that's happened within the web server attempting to get you an ad. It's typically a server-side problem out of your control
kADFAdViewErrorNoAdAvailabe	no ad available in AdFalcon stores
kADFAdViewErrorInvalidParam	There is a parameter has invalid value
kADFAdViewErrorMissingParam	You have missed to fill required parameter
kADFAdViewErrorGenericSDK	An error happened within the SDK during attempting to load or render an ad.
kADFAdViewErrorCommunication	No connection available to the internet.





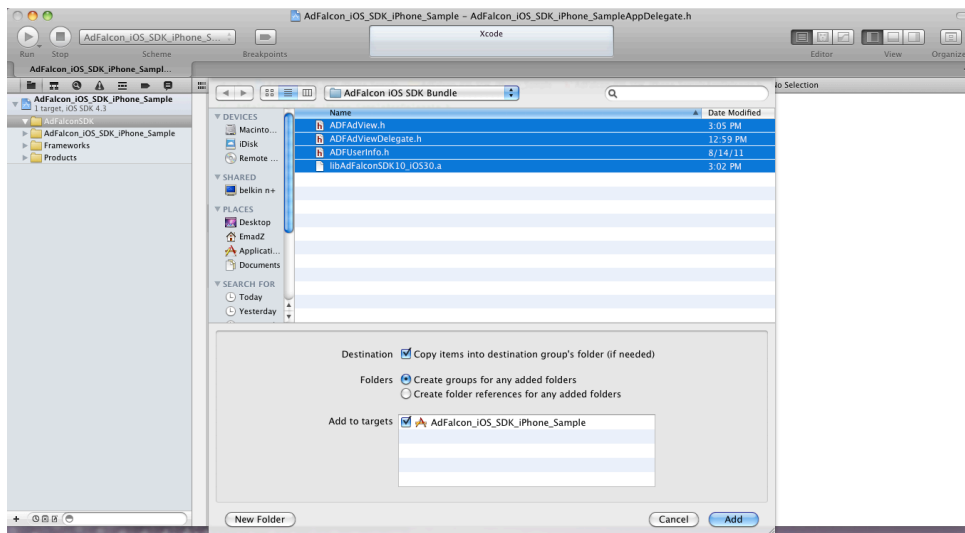
3 AdFalcon iOS SDK Integration

This section describes how to integrate with the AdFalcon's iOS SDK in order to consume the services available by the network.

in order to integrates with the SDK, please perform the following steps:

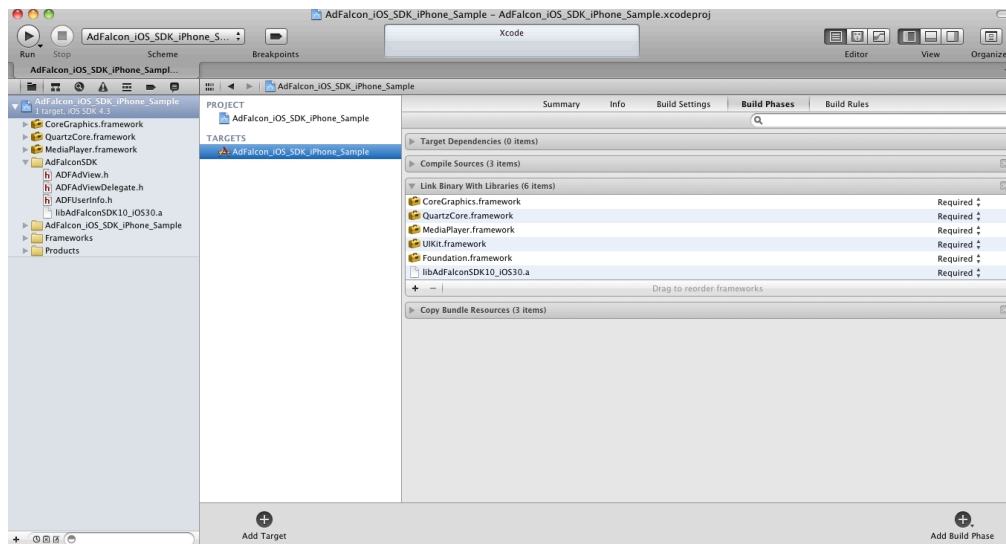
Step 1: Add headers and library

1. Create a folder for AdFalconSDK in your project
2. Right click at the folder and select from submenu Add Files to "Project name"...
3. Navigate to AdFalcon iOS SDK Bundle folder (where the SDK files are stored)
4. Select from the file pane the following headers and library:
 A - **ADFAView.h**
 B - **ADFAViewDelegate.h**
 C - **ADFIUserInfo.h**
 D - **ADFTargetingParams.h**
 E - **libAdFalconSDK1.0.4_iOS30.a**
5. Mark "Copy items into destination group's folder of needed" as checked then click Add



Step 2: Add frameworks

1. Get TARGETS pane and select your target.
2. Select Build Phase's tab.
3. Expand Link Binary With Libraries.
4. click "+" to add the following frameworks to your project
QuartzCore.framework, MediaPlayer.framework, CoreGraphics.framework and SystemConfiguration.framework



Step 3: Add snap of code to a header file

1. Import **ADFAView.h**, **ADFAViewDelegate.h**, **ADFUserInfo.h** and **ADFTargetingParams.h** in your .h file .
2. declare **ADFAViewDelegate** protocol and **ADFAView** instance.

Below is an example of how the header can look like.

```
#import "ADFAViewDelegate.h"
#import "ADFAView.h"
#import "ADFUserInfo.h"
#import "ADFTargetingParams.h"

@interface AdFalcon_iOS_SDK_iPhone_SampleViewController:
    UIViewController<ADFAViewDelegate> {
    ADFAView * adView;
}
```

Step 4: Add snap of codes to a source file

Add the following code in viewDidLoad method

Notes:

- In case you are overriding the `[UIViewController loadView]` to programmatically draw your view, ensure that AdFalcon view is initialized after completing loading the `UIViewController`'s view.
- Ensure Test Mode Parameter is set to false before uploading your application to Apple store.

```
//initialize AdFalcon view and set size of the view base on ad unit size
adView = [[ADFAView alloc] initWithFrame:CGRectMake(0, 0, 320, 48)];
```



```

//Create instance of ADFTargetingParams to store all needed info about user and
application, all this information are optional
ADFTargetingParams * params = [[[ADFTargetingParams alloc] init] autorelease];

//Determine ad keywords i.e. if you set sport, AdFalcon network will retrieve ads
related to sport
//params.keywords = [[NSArray alloc] initWithObjects:@"sport", @"music", nil]
//autorelease];
//params.search = @""; //Not supported

//Create instance of ADFUserInfo
//ADFIUserInfo * user = [[[ADFIUserInfo alloc] init] autorelease];
//params.userInfo = user;
//user.gender = kADFIUserInfoGenderMale;

//This property used to determine a language of ad
//user.language = @"ar";

//Convert NSString date to NSDate
//NSString *dateString = @"21.11.1984";
//NSDateFormatter *dateFormatter = [[[NSDateFormatter alloc] init] autorelease];
//[dateFormatter setDateFormat:@"dd.MM.yyyy"];
//NSDate *dateFromString = [[[NSDate alloc] init] autorelease];
//dateFromString = [dateFormatter dateFromString:dateString];

//Add birthdate or Age
//user.birthdate = dateFromString;
//user.age = 27;

//Location information
//user.countryCode = @"JO";
//user.areaCode = @"962";
//user.postalCode = @"11121";

//If you uses gps location you could pass user location information
//user.locationLatitude = 31.956641;
//user.locationLongitude = 35.847037;
//user.locationAccuracyInMeters = 100;

//If you want to enable logging
//adView.logging = NO;

//If you want to modify refresh duration
//the minimum available duration is 15 seconds
//adView.refreshDuration = 15;

//If you want to use test mode
adView.testing = YES;

//initialize AdFalcon view and loading ads
[adView initWithAdUnit:kADFIAdViewAdUnitSize320x48

//Size of ad if you set a size for adView not match to the Ad unit size; the SDK will
modify it to actual size

        siteId:@"XXXXXXXXXXXXXXXXXXXXXXXXXXXX" //site ID from AdFalcon
web site

        params:params //User information

        rootViewController:self //rootViewController

        enableAutorefresh:YES //if you want to auto refresh for add set this
value to YES other wise you will do this operation manually using refreshAd method

        delegate:self //Optional

];

//Add AdFalcon view
[self.view addSubview:adView];

```



Add the following code in dealloc method

```
[adView release];
```

Add the following optional methods in case you aim to implement the ADFAViewDelegate protocol.

```
-(void)adViewWillLoadAd:(ADFAView*) adView
{
}
-(void)adViewDidLoadAd: (ADFAView*) adView
{
}
-(void)adView: (ADFAView*) adView didFailWithCode:(int) code message:(NSString*)
message
{
}
-(void)adViewWillPresentScreen: (ADFAView*) adView
{
}
-(void)adViewDidPresentScreen: (ADFAView*) adView
{
}
-(void)adViewWillDismissScreen: (ADFAView*) adView
{
}
-(void)adViewDidDismissScreen: (ADFAView*) adView
{
}
-(void)applicationWillTerminate:(UIApplication *)application
{
}
-(void)applicationDidEnterBackground:(UIApplication *)application
{
}
```



4 More Information:

You can find more information in the sample project within the downloaded zip file.

For any SDK integration queries, please send us an email to support@adfalcon.com along with your login id.