



AT&T Advertising SDK Developer Guide for iOS

Publication Date: September 24th 2013



Legal Disclaimer

This document and the information contained herein (collectively, the "**Information**") is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) ("**You**" and "**Your**") by AT&T, on behalf of itself and its affiliates ("**AT&T**") for informational purposes only. AT&T is providing the Information to You because AT&T believes the Information may be useful to You. The Information is provided to You solely on the basis that You will be responsible for making Your own assessments of the Information and are advised to verify all representations, statements and information before using or relying upon any of the Information. Although AT&T has exercised reasonable care in providing the Information to You, AT&T does not warrant the accuracy of the Information and is not responsible for any damages arising from Your use of or reliance upon the Information. You further understand and agree that AT&T in no way represents, and You in no way rely on a belief, that AT&T is providing the Information in accordance with any standard or service (routine, customary or otherwise) related to the consulting, services, hardware or software industries.

AT&T DOES NOT WARRANT THAT THE INFORMATION IS ERROR-FREE. AT&T IS PROVIDING THE INFORMATION TO YOU "AS IS" AND "WITH ALL FAULTS." AT&T DOES NOT WARRANT, BY VIRTUE OF THIS DOCUMENT, OR BY ANY COURSE OF PERFORMANCE, COURSE OF DEALING, USAGE OF TRADE OR ANY COLLATERAL DOCUMENT HEREUNDER OR OTHERWISE, AND HEREBY EXPRESSLY DISCLAIMS, ANY REPRESENTATION OR WARRANTY OF ANY KIND WITH RESPECT TO THE INFORMATION, INCLUDING, WITHOUT LIMITATION, ANY REPRESENTATION OR WARRANTY OF DESIGN, PERFORMANCE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, OR ANY REPRESENTATION OR WARRANTY THAT THE INFORMATION IS APPLICABLE TO OR INTEROPERABLE WITH ANY SYSTEM, DATA, HARDWARE OR SOFTWARE OF ANY KIND. AT&T DISCLAIMS AND IN NO EVENT SHALL BE LIABLE FOR ANY LOSSES OR DAMAGES OF ANY KIND, WHETHER DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL OR EXEMPLARY, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, LOSS OF GOODWILL, COVER, TORTIOUS CONDUCT OR OTHER PECUNIARY LOSS, ARISING OUT OF OR IN ANY WAY RELATED TO THE PROVISION, NON-PROVISION, USE OR NON-USE OF THE INFORMATION, EVEN IF AT&T HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSSES OR DAMAGES.



Table of Contents

Contents

1	Introduction	1
2	About the Advertising API.....	2
2.1	Prerequisites	2
2.2	Setting up your iOS Project	2
2.3	Using the ATTAdView Class.....	5
2.3.1	Configuring ATTAdView Properties.....	5
2.3.2	Initializing the ATTAdView Object.....	6
2.3.3	Implementing Success and Failure Callback Methods.....	7
2.3.4	Advertising API Work Flow.....	8
3	Advertising API Reference.....	9
3.1	ATTAdView Methods	9
3.2	ATTAdView Properties	10
3.2.1	Category Property Values	13
3.3	ATTAdViewDelegate Callbacks.....	15
3.4	ATTAdView Error Codes	16



Table of Figures

Figure 2-1: ATTAdKit folder in the Finder	3
Figure 2-2: ATTAdKit folder in the Xcode Project Hierarchy.....	3
Figure 2-3: Confirmation Dialog.....	4
Figure 2-4: System Frameworks.....	4
Figure 2-5: Xcode Build Settings	5
Figure 2-6: Advertising API Work Flow	8



Table of Tables

Table 3-1: ATTAdView Methods	9
Table 3-2: ATTAdView Properties	13
Table 3-3: ATTAdViewDelegate Callbacks.....	15
Table 3-4: NSError Object Code Property Values	16



Table of Examples

Example 2-1: viewDidLoad Method.....	6
Example 2-2: ATTAdViewDelegate Callback Methods.....	7



1 Introduction

The AT&T Advertising SDK for iOS provides a library, documentation, and sample code that assist developers in building mobile apps using the AT&T Advertising API. The library contains classes and methods for handling authentication and authorization, requests for advertisement content, and rendering the advertising component in the designated frame.

The Advertising API enables your app to support advertisements within the app. This allows the developer of the application to collect a revenue share of the advertisement. When users click the advertisement in the app, they are redirected to the web page for the advertisement.

iOS is the software platform for the Apple iPhone, iPod touch, and iPad. This SDK supports applications that are developed with the iOS Software Development Kit and are distributed by the iOS App Store. The iOS SDK, which is available at <http://developer.apple.com>, provides the tools and APIs that are necessary to develop applications on the iOS platform using the Objective-C programming language. Xcode is available at the Apple developer web site <https://developer.apple.com/xcode/index.php>.

This SDK supports applications that target iOS version 3.1 or higher and Xcode version 4.0 and higher.



2 About the Advertising API

The Advertising API is packaged in the static library ATAdKit. This library exports the ATAdView class, which contains methods that perform the following tasks.

- Handle authentication and authorization.
- Handle click events in the advertising component.
- Get the advertisement.
- Render the advertisement content to the designated area of your app.

The methods in the library contain parameters that allow you to create an advertising component based on filtering criteria and display it in the designated area of your app. The library components require the App Key and App Secret that are created when you register your app for the Advertisement API on the [AT&T Developer Program web site](#).

2.1 Prerequisites

To develop Advertising API apps for iOS, you must have the following prerequisites.

- A Mac OS X computer that is capable of running the Xcode tools. To install the iOS SDK and the Xcode development tools, follow the instructions on the Apple Developer website.
- You must be a member of Apple's iOS Developer Program to run apps on a device and to distribute apps to customers.
- You must understand how to program iOS apps using Objective-C.

2.2 Setting up your iOS Project

To incorporate the Advertising API into your app, perform the following procedure.

1. Open the Xcode project for your app.

Display the unzipped ATAdKit folder in the Finder, as shown in the following illustration.

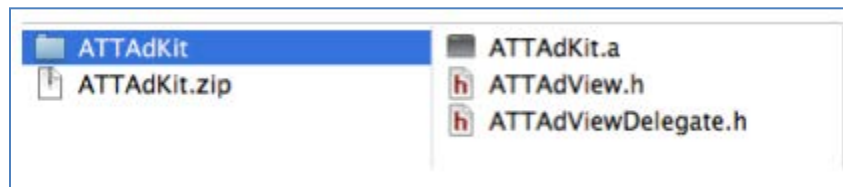


Figure 2-1: ATAdKit folder in the Finder

2. Drag the ATAdKit folder from the Finder to the Xcode project hierarchy, as shown in the following illustration.

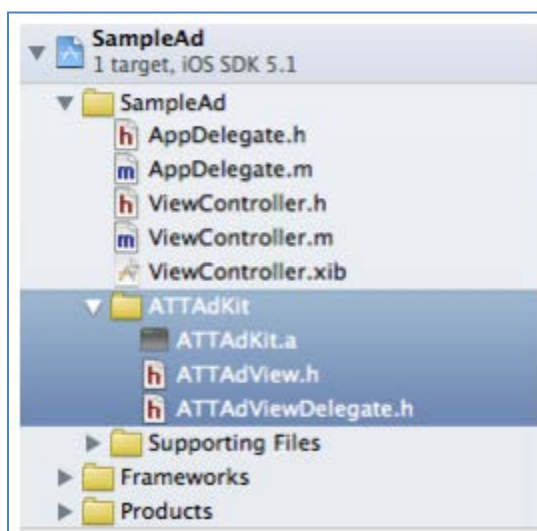


Figure 2-2: ATAdKit folder in the Xcode Project Hierarchy

3. In the confirmation dialog, select the Copy items into destination group's folder (if needed) check box. Also, ensure that your application target is selected in the Add To Targets list. The result should be similar to the following Illustration.

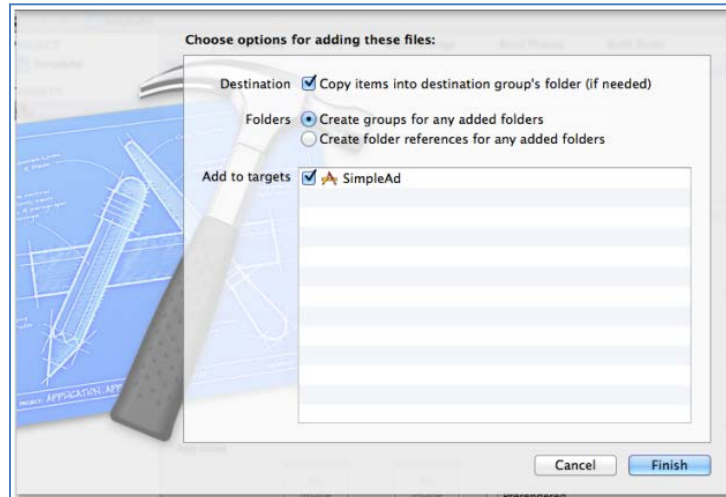


Figure 2-3: Confirmation Dialog

4. Click **Finish**.
5. In your Xcode project, right-click on the Frameworks group and select Add > Existing Frameworks. Select the following system frameworks.
 - SystemConfiguration.framework
 - CoreLocation.framework
 - Security.framework

The result should be similar to the following illustration.

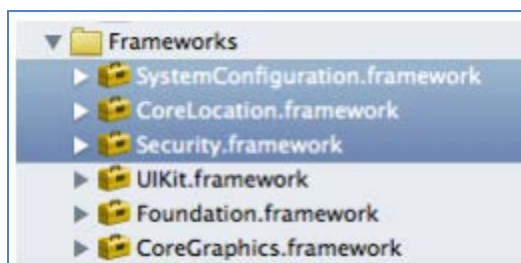


Figure 2-4: System Frameworks

6. Click Add.
7. Set the Other Linker Flags in Xcode Build Settings to ObjC -all_load. The result should be similar to the following illustration.



Figure 2-5: Xcode Build Settings

After you have set up your iOS Project, follow the instructions in [Section 2.3 Using the ATAdView Class](#).

2.3 Using the ATAdView Class

This section describes how to use the ATAdView class in your app, including implementing the listener callback methods.

2.3.1 Configuring ATAdView Properties

You must set values for the following properties of the ATAdView class to authenticate and authorize your app to use the Advertising API.

- appKey
- category
- secret
- udid

You can set additional properties in the ATAdView object to filter the advertisement content. These properties include keywords, city, and zip code. For a full list of properties, see [Section 3.2 ATAdView Properties](#).



2.3.2 Initializing the ATTAdView Object

To initialize an ATTAdView object, insert code similar to the following in the viewDidLoad method.

viewDidLoad Method

```
1 | // Create the ATTAdView object
2 | ATTAdView *attAdView = [[ATTAdView alloc] init];

3 | // Set properties to filter the ad content
4 | [attAdView setZipCode: 98101];
5 | [attAdView setPremium: PremiumBoth];
6 | [attAdView setAgeGroup: @"26-35"];
7 |

8 | // Initialize authentication/authorization, ad service call and
9 | // draw the ad component with the received response
10 | attAdView = [attAdView initWithFrame: CGRectMake(0, 0,
    self.view.frame.size.width, 50) appKey: APP_KEY
    appsecret: APP_SECRET category: @"other" udid: UDID];
11 |
12 | // Add the view
13 | [self.view addSubview: attAdView];
```

Example 2-1: viewDidLoad Method

You must supply an App Key and App Secret value for the APP_KEY and APP_SECRET placeholders in this code, and you must update the Category and UDID properties. Since the App Key and App Secret values provided by AT&T are confidential, you should obfuscate them before passing them to the Advertising service. The sample app in the SDK demonstrates how to obfuscate and un-obfuscate your code.



2.3.3 Implementing Success and Failure Callback Methods

To handle the success or failure when your app requests an advertisement, implement the `ATTAdViewDelegate` callback methods so that your app receives the following callbacks.

- `didReceiveAd:jsonResponse`
- `didFailedToReceiveAd:nsError`

The following code sample shows the signature of these methods with code comments that describe the values returned by the Advertising service that can be used to implement these two delegates.

ATTAdViewDelegate Callback Methods

```
1 | - (void) didReceiveAd: (NSString *) jsonResponse {  
2 | | // The raw JSON response received from the ad service  
3 | | // is available in the variable jsonResponse.  
4 | | }  
5 |  
6 | - (void) didFailedToReceiveAd: (NSError *) nsError {  
7 | | // The error code and message are available  
8 | | // in the variable nsError.  
9 | | }
```

Example 2-2: ATTAdViewDelegate Callback Methods



2.3.4 Advertising API Work Flow

The following illustration shows the work flow of the Advertising API.

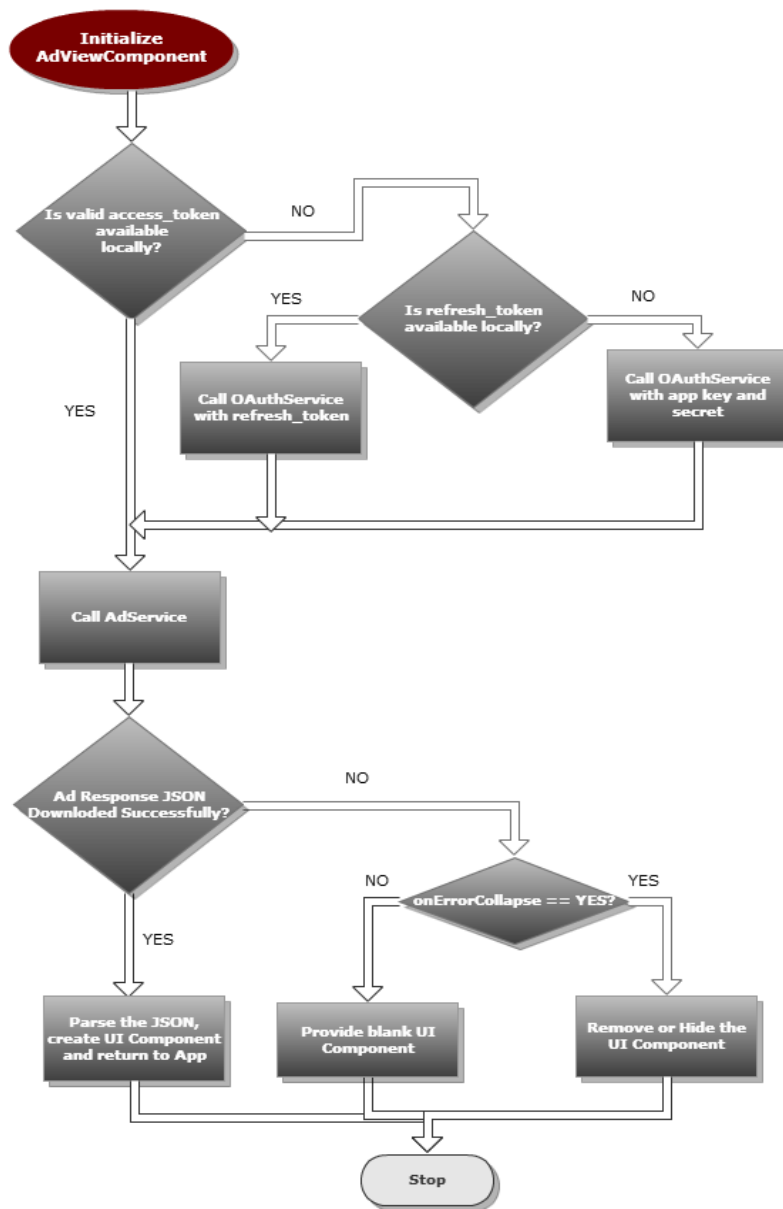


Figure 2-6: Advertising API Work Flow



3 Advertising API Reference

This section provides reference information on the `ATTAdView` and `ATTAdViewDelegate` classes.

- [ATTAdView Methods](#)
- [ATTAdView Properties](#)
- [ATTAdViewDelegate Callbacks](#)
- [ATTAdView Error Codes](#)

3.1 ATTAdView Methods

The following table describes the methods in the `ATTAdView` class.

Method	Result Type	Required/Optional	Description
<code>initWithFrame:appKey:appsecret:category:adCategory</code>	void	Required	This method Initializes the advertisement component and renders it in the specified frame.

Table 3-1: ATTAdView Methods



3.2 ATTAdView Properties

The following table describes the properties in the ATTAdView class.

Property	Default Value	Type	Required/Optional	Description
udid		NSString	Required	Specifies the UDID that is provided by the developer. Must be at least 30 characters in length.
category		NSString	Required	Specifies the category filter for the advertisement. The value of this property filters the advertisement content based on one of the following categories listed in Section 3.2.1 Category Property Values .
gender	nil	NSString	Optional	Indicates the gender of the audience demographic. The defined values for this parameter are M for male and F for female:
zipCode	nil	NSInteger	Optional	Specifies the USA Zip code of the app user.
areaCode	nil	NSInteger	Optional	Specifies the USA area code of app user.
city	nil	NSString	Optional	Specifies the USA city and state of the user.
country	nil	NSString	Optional	Specifies the country of the visitor; it overrides the country detected by IP. It is an ISO 3166 standard that is used for specifying country code.
longitude		double	Optional	Specifies the current longitude, in degrees, of the geographical position for the mobile device.



Property	Default Value	Type	Required/ Optional	Description
latitude		double	Optional	Specifies the current latitude, in degrees, of the geographical position for the mobile device.
maxHeight		NSInteger	Optional	Specifies the maximum height, in pixels, of the advertisement banner. The height of the content may be less than or equal to this value.
maxWidth		NSInteger	Optional	Specifies the maximum width, in pixels, of the advertisement banner. The width of the content may be less than or equal to this value.
minHeight		NSInteger	Optional	Specifies the minimum height, in pixels, of the advertisement banner. The height of the content may be greater than or equal to this value.
minWidth		NSInteger	Optional	Specifies the minimum width, in pixels, of the advertisement banner. The width of the content may be greater than or equal to this value.
timeout	1000	NSInteger	Optional	Specifies the amount of time, in milliseconds, that the user is willing to wait for a response. The maximum value is 3000ms (three seconds) and the default value is 1000ms (one second).



Property	Default Value	Type	Required/ Optional	Description
ageGroup		NSString	Optional	The age group of the demographic audience of the app. The defined values of this parameter are: <ul style="list-style-type: none"> • 1-13 • 14-25 • 26-35 • 36-55 • 55-100
over18	nil	NSInteger	Optional	Specifies whether to display adult ads. The defined values for this parameter are: <ul style="list-style-type: none"> • Do not display adult ads • Show only adult ads. • Show all ads
keywords	nil	NSString	Option	Specifies the keywords that are used to filter the ads. The values are not case-sensitive and multiple values must be separated by commas. For example, to filter for ads about music, tv, or games, use "music,tv,games".
isSizeRequired	No	BOOL	Optional	Indicates whether the size of the advertisement must be returned in the response.
premium	0	NSInteger	Optional	Specifies whether to show premium ads. The valid values for this parameter are: <ul style="list-style-type: none"> • Do not show premium ads • Show only premium ads. • Show all ads
adRefreshPeriod	2	NSInteger	Optional	Specifies the interval, in minutes, at which the advertisement should be refreshed. A value of 0 indicates that the advertisement should not be refreshed.



Property	Default Value	Type	Required/ Optional	Description
onErrorCollapse	No	BOOL	Optional	Indicates whether to collapse the Advertising component, which hides the area from the user if there is any error getting the advertisement.
autoDetectLocation	No	BOOL	Optional	Indicates whether the ATTAdView object should automatically detect the location of the device. When this value is YES, the location is automatically detected and passed to the Advertising service. Note that getting the ad may be delayed if this feature is enabled

Table 3-2: ATTAdView Properties

3.2.1 Category Property Values

The valid values for the Category property include the following.

- Auto
- Business
- Finance
- Chat
- Community
- Social
- Personals
- Communication
- Technology
- Games
- Health
- Medical
- Maps



- Local
- Entertainment
- Movies
- TV
- Music
- Photos
- Video
- News
- Weather
- Shopping
- Sports
- Tools
- Travel
- Other



3.3 ATTAdViewDelegate Callbacks

Once the advertisement is requested, one of the delegate methods of the ATTAdViewDelegate class is called. The delegate may implement any of the methods that communicate the success or failure of loading the advertisement.

The following table describes the delegate callback methods in the Advertising API.

Callback Method	Return	Required/ Optional	Description
didReceiveAd:jsonResponse	void	Optional	This method is invoked when an advertisement is successfully delivered and it renders properly. This method returns a reference to the raw JSON response received from the Advertising service.
didFailedToReceiveAd:error	void	Optional	This method is invoked when authentication fails, an advertisement is not delivered, or the advertisement cannot be rendered. It returns a reference to an NSError object that contains error information.

Table 3-3: ATTAdViewDelegate Callbacks



3.4 ATTAdView Error Codes

When the `adViewDelegate:didFailedToReceiveAd` callback method is invoked, the app can determine what happened by examining the properties of the code property of the `NSError` object. The `adViewDelegate:didFailedToReceiveAd` callback method receives errors from a variety of sources, such as network errors, HTTP error results, and errors reported by the operating system. It follows the standard practice for `NSError` objects, so each source of the error corresponds to a value of the domain property of the `NSError` object and each distinct error from a source corresponds to a value of the code property of the `NSError` object. For advertising service errors, the value of the domain property of the `NSError` object is always `ATTAdErrorDomain`.

The following table describes the possible values of the code property of the `NSError` object.

NSError Code	Description
<code>ATTAdErrorCodeAuthFailure</code>	The Advertising API was unable to authenticate the app with the supplied App Key and App Secret.
<code>ATTAdErrorCodeServiceError</code>	The Advertising API was unable to retrieve an advertisement from the Advertising service.
<code>ATTAdErrorCodeNetworkError</code>	The device was unable to connect to internet.
<code>ATTAdErrorCodeMissingParameter</code>	One or more values for the mandatory properties is either missing or is not valid.
<code>ATTAdErrorCodeOtherError</code>	An error not classified above.

Table 3-4: NSError Object Code Property Values